Instituto Alberto Luiz Coimbra
de Pós-Graduação
e Pesquisa de Engenharia
Universidade Federal
do Rio de Janeiro

Programa de Engenharia de Sistema e Computação

Segurança em redes sem fio

Por
Fernando Verissimo

Monografia entregue à Universidade Federal do Rio de Janeiro, como requisito final do curso de Tópicos Especiais em Redes Integradas Faixa Larga (COS871).

Orientador: Prof. Luís Felipe Magalhães de Moraes

Rio de Janeiro
Janeiro de 2002
v.1.0.1

Agradecimentos

À Academia Brasileira de Ciências, a minha segunda casa.

Agradeço aos colegas Carlos Alberto Vieira Campos, Luís Rodrigo de Oliveira Gonçalves e André Sarmento Barbosa que foram professores.

Ao amigo Marcelo Sávio pelas dicas.

Dedicatória

Dedico esse trabalho aos meus pais, Dionisio e Irene, e a minha esposa, Débora, pelo amor que recebo.

Sumário

Introdução e Motivação

Os avanços da comunicação nos últimos anos possibilitaram o surgimento de várias tecnologias que, desde então, procuram atender a real necessidade de seus usuários, com a melhor qualidade possível. No início eram máquinas mono-usuário, e muito se teve que evoluir até chegar as redes de computadores atuais. Hoje em dia, o mercado está apostando numa das mais novas e revolucionárias tendências tecnológicas: A comunicação por redes sem fio (wireless networks).

A vida do homem pós-moderno é agitada, exige mobilidade, agilidade e liberdade. Esses homens também precisam, cada vez mais, se comunicar onde quer que estejam. Então, os dispositivos de comunicação móvel tornam-se cada vez mais comuns. Os telefones celulares, PDAs, *Notebooks*, entre outros, são dispositivos acessórios que a cada dia são mais comuns. O custo vem caindo a cada ano e alguns modelos de PDA são objetos de promoção de vendas de assinatura de jornais e revistas, distribuídos em grande quantidade para o público em geral.

Já as empresas prestadoras de serviço, que comercializam serviços específicos para dispositivos móveis começaram a surgir só no início do ano passado, e ainda são modestas e prestam serviços básicos, mas só enquanto a

demanda por mais serviços não aumentar. Idéias para novos serviços há muitas.

A tendência mundial é a de criarmos cada vez mais redes mistas, com trechos mais distantes ou de difícil acesso utilizando-se redes sem fio e as redes locais utilizando-se as redes cabeadas. Salvo casos atípicos, onde, por exemplo, uma rede local é instalada em um prédio ou lugar de valor histórico, que, por isso, não se pode passar cabos pelas paredes (já fragilizadas com ação do tempo).

Também se vem falando muito na criação das redes pessoais (PAN), que seriam as redes formadas pelos aparelhos pessoais, como o telefone celular ou o PDA. Essas redes, por serem formadas por aparelhos tão móveis quanto os seus usuários, só fazem sentido se usarem tecnologia sem fio. Entretanto, o diâmetro máximo da rede como essa não ultrapassa os 9 ou 10 metros, devido a limitações tecnológicas.

De outro lado, segurança sempre foi uma preocupação constante do homem. Uma boa definição de segurança temos a seguir:

> *"Segurança são procedimentos para minimizar a vulnerabilidade de bens (qualquer coisa de valor) e recursos, onde vulnerabilidade é qualquer fraqueza que pode ser explorada para violar um sistema ou as informações que ele contém".*
>
> *(Soares1995, p.448).*

Desde o início da humanidade, o homem vem se preocupando com segurança, seja a sua própria segurança, a da sua família ou de seus bens. Nos dias de hoje, a informação é um dos bens mais preciosos que homem possui e

também é um dos mais fáceis de serem perdidos, deturpados ou roubados.

No passado a informação que não ficava somente no cérebro do seu proprietário, era guardada em papéis, que mesmo com o advento da fotocopiadora continuava sendo um meio lento, e por isso, difícil de ser copiado. Agora, a informação é guardada em mídias eletrônicas, onde a cópia é fácil como o processo de arrastar de um mouse. Para aumentar a vulnerabilidade, as informações circulam através de redes, que podem ser locais a uma mesma empresa ou podem ser do tamanho do planeta. Cada vez mais as empresas ligam seus computadores em rede e ligam estas ao mundo exterior, hoje representada pela Internet, em busca de oferecer seus serviços a novos clientes. Cada vez mais cidadãos comuns têm acesso à Internet, fazendo compras com seus cartões de crédito, efetuando operações bancárias, etc...

Por tudo isso, as redes passaram a ser lugares atrativos às pessoas de má fé. A influência do comércio, da indústria, do marketing sobre as redes atrai os *hackers* e *crackers*, quanto um muro recém pintado de branco atrai grafiteiros e pichadores. Aliás, eu costumo utilizar muito essa analogia quando sou perguntado do porquê pessoas sentam-se aos seus computadores, perdendo tempo de suas vidas, para construir vírus, que só servem para causar prejuízos. Costumo responder que ainda estou pensando o porquê pessoas gastam dinheiro em tinta *spray*, arriscam suas vidas escalando marquises, somente com o intuito de colocar uma pichação, que na maioria das vezes é ilegível, em uma pintura limpa de uma casa ou loja. Mas acontece que hoje, os *crackers* ainda me intrigam, assim como os pi-

chadores. Temos que nos preocupar não com pessoas que querem roubar as informações, ou pessoas que querem saber informações sigilosas para poder revendê-las ou fazer chantagens, mas devemos preocupar com pessoas que querem invadir nossos sistemas somente para destruí-lo, não precisando de nenhum retorno financeiro, “só pelo esporte”.

Perigo é o que não falta. Basta dizer que a maioria dos problemas de segurança é causada intencionalmente por pessoas que tentam obter algum benefício ou prejudicar alguém. Os danos causados por um ataque bem sucedido podem causar vários prejuízos. Com os ataques perde-se dinheiro e tempo, além da credibilidade ou imagem do atacado. Quando uma rede é atacada pode-se perder apenas tempo. O tempo de baixar um backup, re-organizar ou re-indexar alguns dados. Entretanto, pode-se perder dinheiro contratando consultoria externa para repor o sistema, ou comprando um sistema de segurança que não era necessário anteriormente. Mas se a empresa trabalha com comércio eletrônico, o pior dos prejuízos é a destruição da imagem da empresa. Qual pessoa voltará a comprar produtos on-line numa empresa, quando soube que essa empresa cuja ele era cliente foi atacada e o sigilo dos números de cartões de crédito foi quebrado?

Por tudo isso, hoje, custa mais caro repor o que foi perdido, do que se proteger.

A minha pesquisa basear-se-á nesses dois tópicos importantes: A segurança de redes de computares e a segurança em redes sem fio de computadores.

No primeiro capítulo vamos falar sobre a segurança de redes de computadores. Mostraremos os tipos de ataques e

os tipos de intrusos. Conversaremos sobre a metodologia de um ataque e como podemos tornar a nossa rede segura. Há um glossário de termos e jargões comuns na área no apêndice *A* deste trabalho.

No segundo capítulo daremos definições sobre métodos de segurança, e veremos a criptografia.

O terceiro capítulo é dedicado a explicar os padrões mais comuns de tecnologia de redes sem fio. São poucos e de fácil entendimento.

No quarto capítulo você lerá sobre o WEP, utilizado como uma camada de segurança em quase todos os padrões de redes sem fio.

O quinto capítulo, e talvez a maior contribuição desta monografia, será destinado a explicar a fragilidade do WEP, defendendo a segurança do algoritmo RC4. Não se incomode se você não entendeu o que eu acabei de dizer, no decorrer desse capítulo, tendo visto o que está no capítulo 3, você passará a entender.

O sexto capítulo é curto e falará sobre a experiência realizada na ilha de Manhattam, na qual se testou a implementação das redes sem fio naquela ilha. O mais importante neste capítulo é o conjunto de sugestões dadas pelos autores daquela experiência.

O sétimo e último capítulo falará dos mecanismos de segurança adicionais encontrados nas implementações CISCO que tem como padrão de segurança o WEP. Veremos também um resumo da teoria sobre *IP Secutiry*, já alvo de outra monografia desse curso [*Mariano2000*], e veremos a aplicações dele em uma *Virtual Privacy Network* (VPN).

No final da monografia sugeriremos temas para futuros trabalhos, que podem ser desenvolvidos por nós ou não, e concluiremos.

Capítulo 1

Uma solução de segurança deve-se levar em consideração o sistema de computação a ser defendido. As soluções "enlatadas", ou seja, as soluções genéricas que são construídas para serem aplicadas a todas as empresas, não são as melhores. As boas soluções são desenvolvidas especialmente para a empresa alvo. Cada empresa tem a sua forma de trabalhar, tem a sua própria equipe e tem a sua metodologia. Não adianta uma solução que vai obrigar uma equipe a usar uma metodologia de trabalho diferente da que já vem usando há 30 anos. Essa metodologia nova tem muitíssima chance de não ser cumprida na sua totalidade. E, na maioria dos casos, uma metodologia que não é cumprida à risca e é tão ineficaz quanto não ter metodologia nenhuma.

Agora, devemos ter em mente que a melhor solução é aquela que é baseada na modelagem de um provável ataque. Devemos pensar como se fossemos *hackers*, tendo assim, uma melhor visualização de todas as falhas do sistema da empresa.

Consideramos um sistema seguro, o sistema que nos traz os seguinte benefícios:

- Privacidade:
- Autenticação

- Integridade
- Não repúdio
- Controle de Acesso
- Disponibilidade

A privacidade nos garante que ninguém não autorizado estará “escutando” o que se está transmitindo na rede. A autenticação garante que a origem da mensagem ou do documento eletrônico foi corretamente identificado, com certeza que a identificação não é falsa. A integridade garante que o que foi transmitido não foi alterado, de forma nenhuma, durante a transmissão. Garante que o que o destinatário recebeu foi exatamente o que o remetente enviou. A não-repudiação consiste no fato de requerer que nem o remetente nem o destinatário de uma mensagem ou de um documento eletrônico sejam capazes de negar a mensagem, nem de negar que tenha sido enviada, nem negar que tenha sido recebida, se realmente isso tenha acontecido. O Controle de Acesso requer que acesso à informação possa ser controlado pela rede que contenha a informação. Há vezes, se quer dar acesso somente de leitura a um arquivo, tem que se garantir que o leitor não pode, de modo nenhum, alterar o conteúdo do que está sendo exibido. E finalmente, a disponibilidade requer que o sistema de computadores esteja disponível para qualquer pessoa autorizada em qualquer momento que ela deseje.

Tipos de intrusos

A tabela 1.1 mostra os tipos de intrusos

Intruso	*Objetivos*
Estudante	Divertir-se bisbilhotando as mensagens de correio eletrônico

	de outras pessoas.
Hacker/Cracker	Testar o sistema de segurança de alguém; ou roubar dados.
Representante de vendas	Tentar representar toda a Europa e não apenas a América
Executivo	Descobrir a estratégia de marketing do concorrente
Ex-funcionário	Vingar-se do ex-empregador
Contador	Desfalcar dinheiro de uma empresa
Corretor de valores	Causar prejuízo para lucrar no valor das ações
Vigarista	Roubar números de cartões de créditos e revendê-los
Espião	Descobrir a força militar do inimigo
Terrorista	Roubar segredos de guerra bacteriológica

Tabela 1.1 – Exemplos dos objetivos de alguns intrusos. [*Tanenbaum1997, p.658*].

Hackers x Crackers

Para facilitar o nosso trabalho daqui por diante, vamos diferenciar os *hackers* dos *crackers*, mas deixando claro que o bom sistema de segurança deve se prevenir contra o ataque dos dois. Sentimo-nos na obrigação de fazer essa separação devido à cobrança da maioria da comunidade on-line. Realmente, o fato de você possuir o conhecimento de como se utilizar uma arma não o torna um assassino.

A mídia, pelo menos a mais leiga, utiliza-se o termo *hacker* para o uso geral. Não tiro sua razão, pois todo *cracker* é um *hacker*. O problema é que nem todo o hacker é um *cracker*, ou seja, nem todo mundo que tem o conhecimento para tentar invadir um sistema de redes, é a pessoa que invade e comete crimes.

Vamos, doravante, identificar o *cracker* como os *hackers* mal-intencionados que invadem por diversão ou para obter vantagens. (*McClure1999, p. xxv*).

O termo *cracker* não é muito utilizado aqui no Brasil, talvez por não ser difundido pela mídia, como vimos, mas talvez por ser confundido com a designação craque, que é dada ao excepcional esportista de um time ou seleção, qua-

se sempre de futebol, ou talvez por receio de ligar o termo *cracker* ao consumidor da droga feita de cocaína.

Também podemos utilizar nessa monografia o termo intruso ou atacante, que possui o mesmo fim.

Ataques

O intruso pode ter quatro comportamentos diferentes em relação às posições da origem e do destino da mensagem. Na figura a seguir, veremos esses comportamentos:

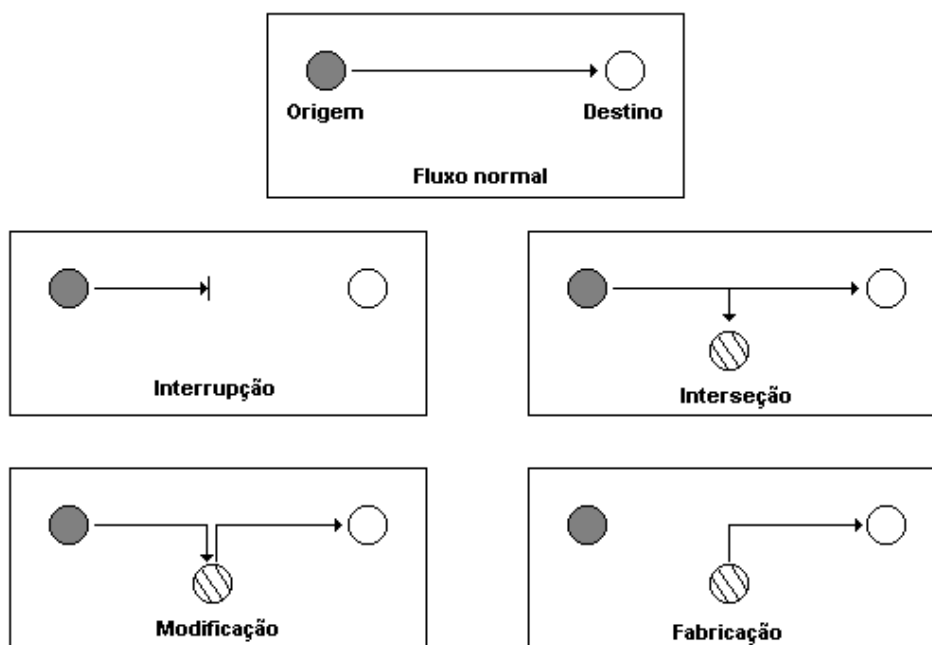



Figura 1.1 – Posição do atacante em relação à origem e ao destino

- **Interrupção:** O intruso objetiva interromper o fluxo de dados que parte da origem, deixando o dispositivo destino sem receber pacotes.
- **Interseção:** Nesse tipo de invasão o intruso objetiva apenas tomar conhecimento de todo fluxo de dados que trafega por essa conexão.

- **Modificação:** Aqui, o intruso além de escutar o tráfego, intercepta os dados e os modifica, enviando-os para o destino.
- **Fabricação:** Na fabricação o intruso fabrica dados para enviar para o destino. O dispositivo destino não tem como saber quem está enviando esses dados.

Adiante vamos falar sobre como são os ataques e os tipos de ataques.

O elo mais fraco

O elo mais fraco de um sistema de segurança é o ser humano. Não tem como se controlar o comportamento de um ser humano. Na frente falaremos sobre a engenharia social, uma técnica utilizada pelo hacker para descobrir as informações necessárias a um ataque.

É comum ouvirmos os especialistas dizerem que o único sistema 100% seguro é aquele que fica o tempo todo desligado. Ora, como um intruso pode invadir um computador desligado? Simples, ele pode pedir para alguém ligar o sistema. Pedir a alguém alguma coisa é uma das ferramentas da Engenharia Social.

Engenharia Social

O termo engenharia social foi dado ao grupo de procedimentos que se toma para convencer alguém a tomar atitudes que você não pode, ou não quer tomar.

A engenharia social é considerada um tipo de ataque a uma rede. Em última instância, pode-se convencer o faxineiro de jogar um balde com água sobre o servidor de email, na hora em que ninguém esteja olhando.

Eu disse em última instância. É lógico que eu imaginei um caso extremo, acho eu que todo faxineiro sabe que circuitos eletrônicos não funcionam muito bem quando estão submersos, porém existem ataques mais brandos que são facilmente encontrados por aí.

Todos esses ataques utilizam pessoas de baixo conhecimento das ameaças que uma rede está submetida ou por pessoas que, por boa fé, querem ajudar. Normalmente são secretárias, estagiários, funcionários novos na empresa, e que querem mostrar serviço.

O mais comum é o intruso, antes de tentar uma invasão, querer saber se a rede tem um *firewall*, qual é esse *firewall*, qual o sistema operacional que roda no roteador, qual o nome, ou o número IP de alguma máquina específica, por exemplo, o servidor de banco de dados. Essas são algumas das informações muito úteis que um intruso pode querer saber antes de um ataque.

Para saber o nome e o telefone do administrador da rede alvo do ataque, pessoa que certamente terá os dados que o atacante quer saber, basta dar uma olhada na internet. Provavelmente esses dados estão na Home Page da empresa que contém a rede. Outro lugar de consulta pode ser a página do Registro.BR.

O Registro.BR é a entidade que controla o registro de nomes de domínios na internet no Brasil. É lá que você registra que o nome xxxx.com.br corresponde à rede 999.999.999.0. Ao registrar essa informação, o administrador da rede deve registrar também alguns de seus dados pessoais. Na maioria esmagadora dos casos, os dados ali registrados são verídicos, mesmo porque, o serviço de re-

gistro de nome de domínio é cobrado por essa entidade (Registro.BR), e ela precisa saber os dados para onde mandar a fatura. Caso os dados estejam errados, a entidade registradora não faturará o serviço e, automaticamente, removerá o registro. Logo, salvo os registros falsos, todos os administradores cadastram seus verdadeiros dados.

Existirá uma possibilidade de, se você telefonar para o administrador, não o encontrar, e em seu lugar, encontrar o seu estagiário. Aí, o intruso pode usar de toda a sua malícia contra o estagiário, que na maioria das vezes, é uma pessoa jovem, sem experiência, e doido para mostrar serviço ao chefe. O intruso liga identificando-se como algum controlador de tráfego do backbone ou de algum órgão do governo, lamenta-se por não encontrar o administrador, e duvida da capacidade do estagiário de lhe dar informações tão específicas, ou seja, desafia o estagiário a mostrar a sua capacidade de dar as informações. Na sua ingenuidade, o estagiário dará todas as informações que ele puder para provar que é capaz, acreditando, assim, ter feito um bom trabalho. Terminará o telefonema feliz, por ter conseguido prestar um bom serviço ao chefe, e deixará o intruso ainda mais contente.

Ex-funcionários

Uma atenção especial deve ser dada à ex-funcionários que saíram contrariados da empresa. Não há como remover os conhecimentos específicos da empresa que foram dados ao ex-funcionário, durante o tempo que ele serviu a essa empresa. Em julho de 2001, o portal de segurança da COPPE/UFRJ, o Lockabit, publicou um artigo sobre esse assunto, onde fala-se sobre a atenção especial que deve

ser dado às informações que são dadas aos funcionários [*Verissimo2001*].

Footprint

Vão existir informações que o intruso não conseguirá coletar através de um telefonema ou um papo amigável com alguma secretária ou estagiário. Seja porque essas pessoas não detêm os conhecimentos necessários, seja porque ele não consegue ter acesso a essas pessoas ingênuas.

Aí, então, surge a segunda técnica de intrusão, conhecida como *footprint*. Consiste em, através de softwares específicos, conseguir informações necessárias ao ataque.

Footprint é um perfil completo da postura de segurança de uma organização que se pretende invadir. Usando uma combinação de ferramentas e técnicas, atacantes podem empregar um fator desconhecido e convertê-lo em um conjunto específico de nomes de domínio, blocos de rede e endereços IP individuais de sistemas conectados diretamente à Internet. Embora haja diversas técnicas diferentes de *footprint*, seu objetivo primário é descobrir informações relacionadas a tecnologias de Internet, acesso remoto e extranet. A Tabela 1.1 mostra essas tecnologias e informações críticas que um atacante tentará identificar.

Tecnologia	***Identifica***
Internet	Nomes de domínio.
	Blocos de rede.
	Endereços IP específicos de sistemas atingíveis via Internet
	Serviços TCP e UDP executados em cada sistema identificado.
	Arquitetura do sistema (por exemplo, SPARC versus X86).
	Mecanismos de controle de acesso e listas de controle de acesso (ACLs, acess control lists) relacionadas.
	Sistemas de detecção de intrusos (IDSs).

	Enumeração de sistemas (nomes de usuários e de grupos, faixas de sistemas, tabelas de roteamento, informações de SNMP).
Intranet	Protocolos de rede em uso (por exemplo: IP, IPX, DexNET, etc...).
	Nomes de domínios internos.
	Blocos de rede.
	Endereços IP específicos de sistemas atingíveis por intermédio da internet.
	Serviços TCP e UDP executados em cada sistema identificado.
	Arquitetura do sistema (por exemplo, SPARC versus X86).
	Mecanismos de controle de acesso e listas de controle de acesso relacionadas.
	Sistemas de detecção de intruso.
	Enumeração de sistemas (nomes de usuários e grupos, faixa de sistemas, tabelas de roteamento, informações de SNMP).
Acesso remoto	Número de telefone analógicos/digitais.
	Tipo de acesso remoto.
	Mecanismo de autenticação.
Extranet	Origem e destino de conexões.
	Tipos de conexão.
	Mecanismos de controle de acesso.

Tabela 1.2 – Tipos de informações procurados num *footprint* [*McClure99, p.6*].

Personificação

Um dos problemas que o intruso encontra quando quer entrar sem permissão em um sistema é a falta de direitos de acesso, e a maneira mais fácil de resolver esse problema é se fazer passar por um outro elemento que tem direitos de acesso ao objeto que o intruso quer invadir.

Depois do *footprint* quase sempre o intruso consegue elementos que identifiquem as pessoas que têm acesso ao objeto alvo. Daí, basta configurar o computador dele com o *login*, nome, número IP que ele deseja personificar.

Replay

No *replay* o intruso intercepta um pacote que vem de um usuário autenticado e reenvia-o novamente mais tarde,

visando confundir os sistemas, ou causando uma parada do sistema.

O sistema que está recebendo os pacotes vai ingenuamente receber os pacotes reenviados pelo intruso, acreditando que ele fora enviado pelo dispositivo origem.

Recusa ou impedimento de serviço

Recusa ou impedimento de serviço, cujo nome em inglês é *Deny of Service (DoS)*, é um ataque muito comum encontrado hoje. Esse ataque consiste no envio de muitos pacotes pelo intruso para um computador. Esse envio torna-se perigoso quando o número de pacotes é muito maior do que a quantidade que o computador atacado pode tratar.

Uma variação mais perigosa é o Impedimento de Serviço Distribuído. Aqui o intruso utiliza-se de outros computadores, conhecidos como computadores zumbis, para aumentar a carga de pacotes (*flood*) a serem tratados pelo computador atacado.

Armadilhas

Também conhecido como *trapdoor* ou *backdoor*. Ocorre quando uma entidade do sistema é modificada para produzir efeitos não autorizados em resposta a um comando (emitido pelo intruso) ou a um evento predeterminado.

Como exemplo, citamos a modificação de um processo para dispensar a verificação de senha na autenticação de um acesso, em resposta a uma combinação de teclas (C-trl+Alt+U) ou a um evento do tipo “hora do sistema =

2:35:00” quando o acesso a qualquer usuário teria a necessidade de senha para autenticação dispensada.

Script Kiddies

Há um tipo de intruso que traz muito perigo. Perigo não por causa de seus conhecimentos avançados, mas por causa da sua aleatoriedade. Os usuários dos *script kiddies* quase todos são hackers iniciantes (algumas vezes, crianças, daí o nome), não tendo ainda conhecimento e experiência suficiente para fazer os seus próprios ataques, e por isso utilizam *scripts* feitos por outros hackers.

O principal problema é que um hacker experiente escolhe as suas vítimas, normalmente são empresas grandes e importantes que estão mais expostos ao grande público, os *scripts kiddies* escolhem suas vítimas ao acaso.

Resumo do capítulo

Uma solução de segurança barata pode sair mais cara. Além dos dados, cuja perda traria um prejuízo incalculável, tem os custos com a recuperação de uma imagem arranhada por uma invasão, que traz sempre a idéia de descuido e falta de compromisso, e os custos trazidos pela falta de vendas, e outros.

Existem soluções de segurança que podem custar até 100 mil dólares, mas não se pode esquecer que esse é o valor do faturamento de alguns dias em algumas empresas de comércio on-line na internet.

Também não existe a segurança perfeita. Todas as soluções tentam dificultar ao máximo uma invasão. Os *hackers* são pessoas muito inteligentes, e continuam pensando em métodos de burlar aquele mais moderno sistema de segurança que você terminou de implementar.

Não confie totalmente em soluções vendidas em pacotes fechados. Normalmente essas soluções são mais baratas, porque são vendidas em grande quantidade, mas são soluções que não te deixarão seguro. Normalmente essas soluções não envolvem o treinamento do pessoal, a conscientização dos funcionários da sua empresa e não é modelada ao tipo de serviço que você presta na internet, o tipo de pessoas que precisam ter acesso remoto à sua rede.

Graças aos *scripts kiddies* e aos DDoS, a sua rede está constantemente sendo atacada. Os *script kiddies* podem a escolher aleatoriamente, e os ataques DDoS a utilizam como zumbi para o ataque a uma outra rede.

Lembre-se, para nos defender de um *hacker*, devemos pensar como um. Pensar em como faríamos para atacar a nossa própria rede, ajudaria a nos defender melhor, pois poderíamos descobrir as vulnerabilidades do sistema antes que o próprio *hacker* descubra-os.

No próximo capítulo veremos como funciona a criptografia, que é a forma de nos defendermos do *hacker* que fica na espreita tentando escutar o que transmitimos pela rede. A principal dica para manter-se segurança é codificar tudo o que é transmitido.

Os especialistas sugerem que você seja um paranóico por segurança. As firmas grandes, que perdem muito com os ataques, possuem pessoas ou equipes cujo trabalho é

cuidar da segurança, instalando novas versões de softwares, treinando e orientando as pessoas que possam se alvos dos ataques de engenharia social, desconfiando de tudo e de todos.

Capítulo 2

Nesse capítulo veremos o que é criptografia. Criptografia, ou algoritmos criptográficos, basicamente objetivam "esconder" informações sigilosas que qualquer pessoa desautorizada possa ler, isto é, de qualquer pessoa que não conheça a chamada chave secreta de criptografia. Criptografia é a ciência de se comunicar secretamente.

Dizem que a criptografia é tão antiga quanto à própria escrita. Desde que surgiu a necessidade de passar, de forma escrita, uma informação para outra pessoa, surgiu a necessidade de passá-la *só* para uma pessoa. O homem a utiliza desde cedo, logo quando ainda é uma criança. As crianças inventam a língua do P e outras formas de brincadeira que ajudam a exercitar o entendimento dessa ciência.

Terminologia

Para ajudar o entendimento deste capítulo, e dos demais, vamos combinar uma terminologia.

Mensagem ou *texto* é à parte da informação que se deseja proteger ou esconder. Um dos principais objetivos da criptografia é garantir o armazenamento e circulação segura de mensagens. Quando se usar o termo *remetente*, está se referindo à pessoa que manda uma mensagem para outra, a qual será o *destinatário*. Na criptografia existem dois tipos de textos (ou mensagens). O primeiro é a mensa-

gem a ser transmitida, na sua forma original. Esta será chamada de *texto puro*. O texto puro passa por um processo que se denomina *encriptação*, e assumirá uma nova forma. Esta nova forma, chamada de *texto cifrado*, é a que será transmitida, e que, quando interceptada por um terceiro, deverá permanecer inteligível. Ao receber o texto cifrado, o destinatário usará um processo que se chama *desencriptação* para recuperar o texto puro. *Criptografar* ou *encriptar* é o ato de encriptação de uma mensagem e *descriptografar* ou *desencriptar* é o ato de desencriptação da mensagem. A *chave* é uma informação que o remetente e o destinatário possuem, e que será usada para criptografar e para descriptografar. Nem sempre a chave do remetente será igual à do destinatário, mas isso nós veremos adiante.

Se a criptografia é usada então os dados a serem criptografados deverão transitar por um *meio*, ou *canal*, não seguro, que pode ser desde o lombo de um cavalo até a Internet. Uma outra possibilidade é a de os dados terem de ser armazenados em um local onde terceiros possam ter acesso. Neste caso, o meio por onde a mensagem "transita" é o local de armazenamento. Já foram apresentados dois personagens: o remetente e o destinatário. Existe também um terceiro personagem: o intruso ou *inimigo*. Este ganha acesso ao texto cifrado ou a algum outro tipo de informação, e tentará fazer uma *análise criptográfica* para tentar recuperar algum tipo de informação a respeito do texto puro (ou o próprio texto cifrado). O inimigo pode adotar também outros tipos de ataques. A análise criptográfica consiste na utilização de alguma técnica, ou técnicas, para obter informações a partir do texto interceptado.

Esse tópico foi escrito em concordância com *[De Carvalho2000, p.1-2].*

Conceitos Matemáticos

Esse tópico é destinado a quem não ainda não possui conceitos específicos, úteis no estudo da criptografia, ou para aqueles que, com o passar dos anos, precisam reviver esses conceitos. São eles:

Bases numéricas

Será bastante comum encontrar números que não estão na base 10, que normalmente conhecemos, ou seja, números que para ser representados utilizam 10 algarismos. Os números que veremos nos nossos estudos podem estar na base 2, graças ao bit, ou na base 16.

Os números na base 2 só utilizam dois algarismos: o 0 e o 1. Os números na base 16 só utilizam 16 algarismos: de 0 a 9 e de A a F.

Operações lógicas

Operações lógicas são operações com variáveis binárias (de base 2), cujos valores podem ser 0 ou 1, ou verdade ou mentira. Sejam *a* e *b* números binários. Serão usadas as operações lógicas mostradas a seguir:

NOT *a* – O valor inverso do bit. Se o valor é 0, ele se transformará em 1, e se o valor for 1, ele se tornará 0. ($\neg$).

a OR *b* – O resultado será 0 se os dois operandos são 0, e 1 nos outros casos. ($\vee$)

a AND *b* – O resultado será 1 se os dois operandos são 1, e 0 nos outros casos. ($\wedge$)

a XOR *b* – O resultado será 0 se os dois operandos forem iguais, e 1 se eles forem diferentes. (⊗)

a	b	¬a	a∨b	a∧b	a⊗b
0	0	1	0	0	0
0	1	1	1	0	1
1	0	0	1	0	1
1	1	0	1	1	0

Tabela 2.1 – Operações lógicas.

A propriedade mais importante da operação XOR é:

$$a \otimes b \otimes a = b \ \forall \ a,b.$$

A importância para a criptografia desta propriedade está no fato de que ela mostra que a operação XOR é *reversível*. Este é o motivo pelo qual a operação XOR é muito utilizada em algoritmos criptográficos.

Quando se faz operações lógicas com variáveis inteiras se está, na verdade, fazendo as operações bit a bit. Isto significa que cada bit da variável *a* passa pela operação com o bit correspondente da variável *b*. Por exemplo, sejam dois registradores de 8 bits *a*=1001.1100 e *b*=0101.1010:

$\neg a$ = 0110.0011

$a \vee b$ = 1101.1110

$a \wedge b$ = 0001.1000

$a \otimes b$ = 1100.0110

Outra operação lógica, de extrema importância na criptografia, é a *rotação*. A rotação desloca os bits de uma variável para a direita ou para a esquerda. Há dos tipos de rotação: a circular e a não-circular. A rotação circular considera a variável como sendo conectada nas extremidades, de maneira que um bit saído de um lado da variável retorna ao outro lado. Na rotação não-circular um bit saindo da va-

riável é eliminado e zeros são colocados nas posições vagas.

Representa-se a rotação para a direita não-circular pelo símbolo ». A rotação não-circular para a esquerda será «. A rotação circular para a direita será ➢ e para a esquerda será ≺. À esquerda do símbolo coloca-se o registrador a ser modificado. À direita do símbolo coloca-se o número de bits a serem deslocado. Alguns exemplos ilustrarão este conceito. Para *a*=0100.1101, tem-se que:

a«3 = 0110.1000

a»2 = 0001.0011

a≺1 = 1001.1010

a➢5 = 0110.1010

<u>Módulo</u>

Uma operação de extrema importância na criptografia é a operação de *redução modular*. Define-se como:

$$x \bmod m = x - (\lfloor x/m \rfloor . m)$$

onde *x*, *m* e *x mod m* são inteiros, e *m>0*. Note-se que sempre *0≤(x mod m)≤(m-1)*. Pode-se dizer também que, no caso de *x>0*, *x mod m* é o resto da divisão inteira de *x* por *m*.

Para simplificar, a operação de redução modular também será chamado de *módulo*, apesar de esta denominação ser mais apropriadamente aplicada ao número *m* usado na redução.

Histórico

Quase todas as pessoas, antes de estudar um pouco de criptografia, acham que o uso de um sistema simples qualquer, bastará para garantir a sua segurança. Um algoritmo de substituição alfanumérica não basta para garantir privacidade ou integridade, por exemplo. Não se esqueçam que o que pode ser complicado para o célebro humano, que tem velocidade limitadamente baixa, não é complicado para um computador que faz milhões de operações por segundo. Aquele software que possui um sistema de criptografia fraco é pior do que aquele outro que não possui criptografia nenhuma. Esse sistema com criptografia fraca cria a expectativa no usuário de segurança. Esse passa a se despreocupar com segurança, baixa as suas defesas, e provavelmente terá as suas informações violadas e/ou destruídas.

Os sistemas antigos que vamos apresentar logo em seguida não foram feitos para serem usados por computadores, ou melhor, ainda, não foram feitos para proteger as informações de ataques comandados por computadores. Esses sistemas foram utilizados para criptografar o alfabeto comum, com 26 letras. Portanto, nesses sistemas antigos adotaremos a convenção que os algoritmos criptografarão letras e não bits ou bytes, quando estivermos falando dos algoritmos mais modernos passaremos a citar bits, bytes e palavras.

A maior parte dos sistemas antigos pode ser reunida em três grupos de técnicas. O primeiro grupo é o das substituições monoalfabéticas. O segundo grupo são sistemas

de permuta. O terceiro grupo é constituído de sistemas de estenografia[1].

Um exemplo clássico da estenografia antiga é o seguinte: Para se transmitir uma mensagem secreta a um aliado, um comandante raspava a cabeça de algum escravo e tatuava a mensagem no seu couro cabeludo. Após o cabelo do escravo ter crescido de novo, este era enviado ao destinatário. Ao chegar lá, o seu cabelo era novamente raspado para se poder ler a mensagem.

Cæsar e os sistemas monoalfabéticos

A característica dos sistemas de substituições monoalfabéticas é que cada letra da mensagem é substituída por uma outra, de tal maneira que esta relação de substituição seja fixa.

A invenção do sistema Cæsar é atribuída ao imperados Júlio César. Neste caso, a chave *K* é um número inteiro entre 0 e 25. Cada letra *l* da mensagem é encriptada usando-se a seguinte equação:

$$e = (l + K) \bmod 26$$

Para desencriptar, usa-se:

$$l = (e + K) \bmod 26$$

Neste caso, logicamente, a chave *K* tem que ser a mesma nas duas operações: encriptação e desencriptação.

A limitação deste sistema é que ele só tem 25 chaves possíveis, mesmo sem o uso do computador não demoraríamos muito para testar as possíveis chaves.

[1] Estenografia é a capacidade de esconder mensagens secretas em um meio, de maneira que as mesmas passem despercebidas. Um exemplo poderia ser escrever uma carta com tinta invisível.

Sistema de permuta

Neste tipo de sistema a chave *K* é um vetor de permuta *P* de tamanho *m* (*m*>=2). O vetor *P* deverá ter todos os números entre 1 e *m*, sem a repetição de nenhum. Estas características garantem que *P* terá uma permuta inversa.

A mensagem a ser criptografada deve ser dividida em blocos de *m* letras. Seja o bloco de texto puro conhecido como *tp*:

$$tp = (a_1, a_2, a_3, ..., a_m)$$

A saída desse algoritmo para este bloco, será um outro bloco de mesmo tamanho, cuja permutação dependerá da chave *K*.

Exemplo:

M=VAMOSATACARAMANHASEMFALTA

P=(3,5,4,1,2)

M'=MSOVAAACATMNARASMEHALATFA

Vigenère e os sistemas polialfabéticos

Sistemas polialfabéticos são aqueles em que se tem a combinação ordenada de diversos sistemas monoalfabéticos. O sistema Vigenère é um sistema polialfabético. Ele adota como chave um conjunto de *p* letras:

$$ch = (l_1, l_2, l_3, ..., l_p)$$

A mensagem deve então ser dividida em blocos de *p* letras. Chama-se *p* de período do sistema polialfabético. Seja um bloco de texto puro *tp*:

$$tp = (a_1, a_2, a_3, ..., a_p)$$

A saída desse algoritmo será um outro bloco, também com *p* letras. Este será a substituição de *tp* usando *ch*:

$$tc = ((a_1 + l_1) \bmod 26, (a_2 + l_2) \bmod 26, (a_3 + l_3) \bmod 26, \ldots, (a_p + l_p) \bmod 26)$$

Pode-se generalizar este conceito. Dado um período *p*, uma chave para uma substituição polialfabética é constituída de *p* chaves para substituições monoalfabéticas. Divi-di-se sempre a mensagem em blocos de *p* letras e, em cada letra de um bloco, aplica-se uma das substituições monoalfabéticas da chave. Sob este ponto de vista, o sistema Vigenère se constitui de *p* substituições monoalfabéticas consecutivas do tipo Cæsar.

É interessante notar que o sistema *One-Time Pad*, que falaremos mais adiante neste capítulo, poderia ser visto como um sistema Vigenère com um período do mesmo tamanho da mensagem.

Sistemas de criptografia modernos

Os sistemas que vimos até agora são fáceis de ser quebrados pelo computador, esse elemento que, comparado com a história da criptografia, é muito recente. Hoje se tem, à mão, um poder computacional impensável há algumas décadas. Com o computador pode-se quebrar facilmente qualquer um dos sistemas citados até o momento.

Com os computadores, as mensagens agora podem representar qualquer coisa, pois, na representação binária dos computadores pode-se ter textos, imagens, programas e sons, entre outros. Os algoritmos são projetados para operar em bits, ou em conjunto de bits, não mais em letras como antes.

Os novos algoritmos inventados para operar com bits, não podem ser mais operados manualmente, pois envolvem um número muito grande de operações. Como o computador permite a comunicação quase instantânea à distância, novos problemas começaram a surgir, como por exemplo, a distribuição das chaves.

Mas antes de começar a falar sobre os modernos algoritmos de encriptação, vamos dar uma olhada em alguns conceitos importantes sobre a desordem de dados e a teoria da informação.

Teoria da Informação

Difusão e confusão

Diz-se que uma substituição acrescenta *confusão* à informação, e uma transposição acrescenta *difusão*. O objetivo da confusão é tornar mais complexa a relação entre a chave e o texto ilegível de tal forma que fique difícil a um criptanalista[2] deduzir qualquer propriedade da chave a partir do conhecimento do texto ilegível.

O objetivo da difusão é embaralhar ou espalhar os bits do texto legível para que qualquer redundância seja eliminada no texto ilegível.

Entropia

A definição de entropia é clara e pode ser encontrada no dicionário:

> ***Entropia****. S. f.* ***1****. ...* ***2****. Medida da quantidade de desordem dum sistema.*
>
> *[Aurelio1986, p.667].*

[2] Pessoa que analisando o texto criptografado tenta descobrir o texto legível utilizando técnicas de análise de criptografia.

A entropia é uma medida e possui a sua fórmula de cálculo. A entropia visa medir quanto os dados estão bagunçados. Quanto mais desordem tiver os dados cifrados, mais complicado será para descobrir uma correlação entre eles e os dados puros.

Definição: Dadas *n* informações $X = \{x_1, x_2, \ldots, x_n\}$ ocorrendo respectivamente com probabilidades $p(x_1), p(x_2), \ldots, p(x_n)$, a entropia é definida pela fórmula:

$$E(X) = \sum_{j=1}^{n} p(x_j) \log_2 \left[\frac{1}{p(x_j)} \right]$$

Como $\log_2[1/p(x_j)]$ representa o número de bits para codificar x_j, tem-se que E(X) é o número médio de bits para codificar todas as informações em X.

Imaginando X com dois elementos, a probabilidade do primeiro elemento igual a *p* e a do segundo elemento igual a *q*, onde, obviamente, *p=1-q*.

E(X)=-(*p*log*p* + *q*log*q*)

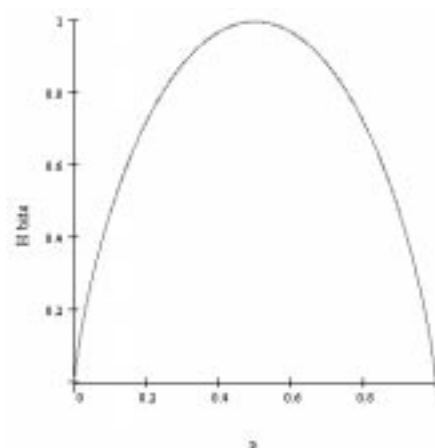



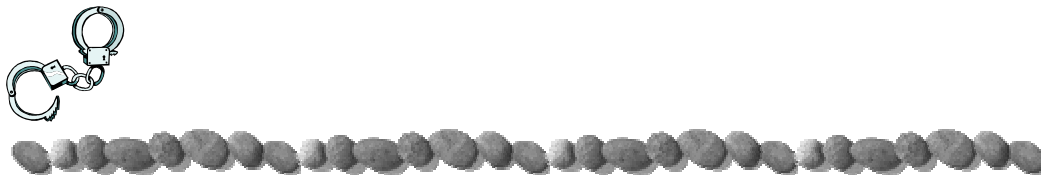
Figura 2.1 – Gráfico de entropia com 2 elementos.

Em 1949, Claude Shannon publicou um artigo seminal chamado Comunication Theory of Secrecy Systems, no Bell

Systems Technical Journal, que influenciou substancialmente o estudo da criptografia. Recomendo aos que tem boa base matemática uma lida na parte do artigo que fala sobre entropia, senão no artigo inteiro. No apêndice A daquele artigo, Shannon prova esta fórmula.

Segurança perfeita

Informalmente, o que o projetista de um algoritmo criptográfico objetiva é que o criptanalista (o intruso) não seja capaz de obter nenhuma informação sobre o texto ilegível se conseguir interceptar o texto ilegível correspondente. Este objetivo chama-se segurança perfeita. Esse termo é conflitante com que já dissemos anteriormente, já que não existe uma segurança perfeita, mas esse é o termo encontrado na literatura, gostemos ou não.

Para formalizar este conceito, é necessário usar conceitos de probabilidade.

Seja $X = \{x_1, x_2, \ldots, x_n\}$ o conjunto finito de *n* textos legíveis, seja $Y = \{y_1, y_2, \ldots, y_u\}$ o conjunto finito de *u* textos ilegíveis, e seja $K = \{K_1, K_2, \ldots, K_r\}$ o conjunto finito de *r* chaves. Cada chave K_s define uma função criptográfica $f_{K_s}(x_i) = y_j$.

Vejamos um exemplo para *n*=2, *u*=4, *r*=3.

Textos legíveis	Chave usada	Textos ilegíveis
x1	K1	y1
x2	K1	y2
x1	K2	y2
x2	K2	y3
x1	K3	y3
x2	K3	y4

Tabela 2.2 – Relação entre textos puros e cifrados na segurança perfeita.

Por definição, o algoritmo terá segurança perfeita se e só se:

$p_j(x_i) = p(x_i)$

Onde

1. $p(x_i)$ é a probabilidade de alguém criptografar o legível x_i
2. $p_j(x_i)$ é a probabilidade de alguém criptografar o legível x_i, dado que y_j foi interceptado pelo criptanalista.

Chaves igualmente prováveis

Como o próprio nome já explica, é o caso em que a probabilidade de se escolher qualquer uma das chaves é igual à probabilidade de escolha de qualquer outra, ou seja, igual a 1/r.

Condição necessária e suficiente para segurança perfeita

Vamos considerar um algoritmo C para o qual o número de textos legíveis *n*, o número de chaves *r* e o número de ilegíveis *u* são iguais. Nestas condições C possui segurança perfeita se e só se:

1. fixado um para (*x*,*y*) de legíveis *x* e ilegíveis *y*, existe exatamente uma chave que criptografa *x* para *y*;
2. e todas as chaves são igualmente prováveis.

One-time-pad

Um algoritmo que satisfaz estas condições de segurança perfeita é chamado de one-time-pad. Supõe-se para este algoritmo que:

1. Exista um limite superior fixo *L* para o comprimento de um texto legível qualquer a ser criptografado.
2. O número total de chaves é maior ou igual a *L*.
3. Todas as chaves são igualmente prováveis.
4. Para criptografar um legível $x = \{x_1, x_2, ..., x_n\}$, cada símbolo k_j da chave $K = \{k_1, k_2, ..., k_n\}$ (no caso extremo, cada bit da chave) é escolhido aleatoriamente e independentemente dos outros símbolos de *K*.
5. O número de escolhas de cada k_j é igual ao número de escolhas de cada x_j.

Nestas condições, o One-time-pad é apenas uma operação que soma cada x_j a k_j módulo *A*, sendo *A* o número total de símbolos.

Infelizmente, o One-time-pad é difícil de ser implementado, pois não se sabe construir um algoritmo gerado de chaves realmente aleatórias. Os algoritmos conhecidos geram números que são apenas pseudo-aleatórios.

<u>Criptossistema aleatório</u>

Para definir um criptossistema aleatório, precisamos primeiramente ver algumas definições básicas.

Dado um alfabeto finito *A* tal que |*A*|=*a*, para qualquer inteiro *n*>0 seja X_n o conjunto de textos de comprimento *n*, e portanto, $|X_n| = a^n$.

A primeira propriedade de um criptossistema aleatório *S* é que o *número de textos legíveis de comprimento n que é* a^n *é igual ao número de textos ilegíveis de tamanho n.*

Todas as chaves do criptossistema aleatório são igualmente prováveis, e por último, o conjunto de textos legíveis pertencentes a *X* tal que através de um conjunto de chaves pertencentes a *K* que são criptografados em um texto ilegível *y* qualquer, deve ser um conjunto aleatório.

Tipos de Criptografia

Existem dois tipos de criptografia, classificados quanto à forma como se escolhe as chaves: A criptografia de chave[3] simétrica ou chave secreta, na qual a chave de criptografar é igual à chave de descriptografar e a criptografia de chave assimétrica, na qual as chaves são distintas. Existem também alguns algoritmos que utilizam uma combinação dos dois tipos de criptografia, e são chamados de algoritmos híbridos.

Nesses algoritmos o que deve ser secreto são as chaves de criptografia e descriptografia. O próprio algoritmo é uma coisa que deve ser amplamente difundido, mesmo porque a ciência ainda não conseguiu desenvolver um método para testar eficientemente os algoritmos. Não existe ainda um método matemático que prove a segurança de algoritmo, logo a melhor maneira de se dizer que um algoritmo tem segurança ou não é expondo-o a prova dos criptanalistas. Para isso, os inventores expõem os seus algoritmos nas feiras e congressos ligados a ciência da criptografia.

[3] Nem sempre uma chave é uma senha. Existem os dispositivos biométricos que aceitam chaves como sendo identificadores pessoais, como exemplo: retina, íris, impressão digital, voz, etc...

Criptografia simétrica

Nessa sessão vamos estudar alguns algoritmos de criptografia de chave simétrica.

A figura a seguir mostrará como funciona um sistema de criptografia com chave simétrica.

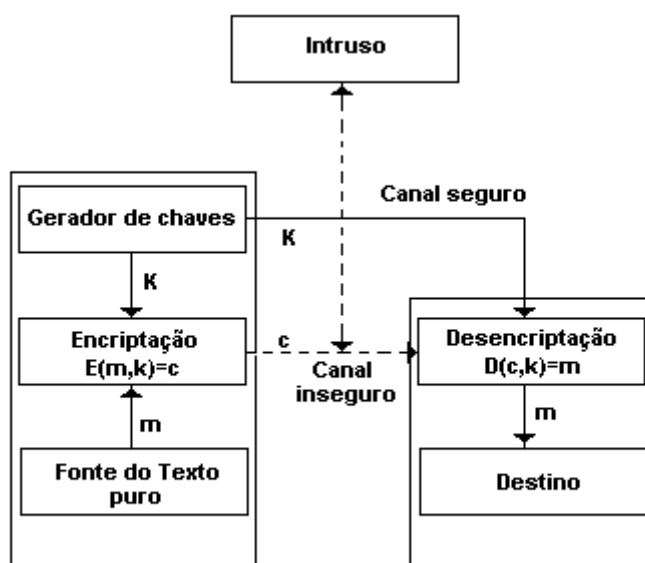



Figura 2.2– Esquema da criptografia chave secreta

Onde:

- *E* é o algoritmo de encriptação
- *D* é o algoritmo de desencriptação
- *K* é a chave secreta
- *m* é o texto puro
- *c* é o texto criptografado

De todos os algoritmos de chave secreta, o mais importante é o *Data Encryption Standard* (DES). Este algoritmo é o mais amplamente usado internacionalmente ainda hoje, e foi um avanço científico significativo no sentido de ter sido o primeiro algoritmo de criptografia cujo conheci-

mento se tornou público: Até então todos os algoritmos e-ram secretos. Ou seja, a segurança do DES não se baseia no conhecimento do algoritmo, mas apenas no conhecimento da chave secreta. O DES foi projetado pela IBM e publicado no *National Bureau of Standards* (NBS) em 1977 para se adotado como padrão nos EUA para informações comerciais.

Entretanto, não é alvo dessa monografia falar sobre todos os algoritmos de criptografia. O único que vai interessar para esse trabalho é o RC4. Veremos no capítulo 4 que o RC4 é algoritmo utilizado pelo WEP.

<u>RC4</u>

O RC4 (*Ron's cipher 4*) foi criado por Ronald Rivest em 1987, e foi mantido em sigilo. O algoritmo era propriedade da RSA Security. Em 1994, alguém enviou, anonimamente, para uma lista de discussão da Internet, um algoritmo, afirmando que aquele era o RC4. Este fato foi confirmado empiricamente, e o que era segredo industrial virou domínio público, da noite para o dia. Felizmente, como um bom algoritmo criptográfico, o RC4 é seguro independente de seu algoritmo ser público ou não.

O RC4 é, na verdade, uma maneira de se gerar bytes aleatórios, a partir de uma chave de tamanho variável. Estes bytes serão usados para encriptar uma mensagem através da operação lógica XOR. O destinatário executará o RC4 como o remetente, obtendo os mesmos bytes aleatórios, podendo assim desencriptar a mensagem.

A principal vantagem do RC4 é que ele é um algoritmo de fluxo. O algoritmo de fluxo é chamado assim, pois con-

catena a string gerada com a mensagem pura à medida que esta última é gerada.

A expansão da chave (KSA)

O RC4 recebe uma chave *ch* de n_{ch} bits, onde 1≤ n_{ch} ≤2048. Tem-se que gerar um vetor *S* de 256 bytes, a partir da chave:

$$S = (s_0, s_1, s_2, ..., s_{255})$$

Para tanto, utiliza-se o seguinte algoritmo:

1. Para *i* de 0 a 255 faz-se
   a. $s_i := i$
2. Seja o vetor de 256 bytes (2048 bits) $K = (k_0, k_1, ..., k_{255})$
3. Copia-se a chave *ch* para *K* bit a bit, repetindo-a quantas vezes forem necessárias para preencher *K* completamente. Por exemplo, se n_{ch}=100 copia-se a chave 20 vezes para *K*, e ainda se coloca os 48 primeiros bits de *ch* no fim de *K* para terminar de preenchê-lo.
4. *j:=0*
5. Seja *t* um byte.
6. Para *i* de 0 a 255 faz-se:
   a. $j := (j + s_i + k_i) \bmod 256$
   b. $t := s_i$
   c. $s_i := s_j$
   d. $s_j := t$

Pode-se perceber que *S* é, de fato, uma permuta dos números de 0 a 255 determinada pela chave.

O algoritmo do RC4 (PRGA)

Para gerar os bytes aleatórios tem-se o seguinte algoritmo:

1. *i:=0*
2. *j:=0*
3. Seja *t* um byte
4. Enquanto foram necessários bytes *b* aleatórios faz-se:
    a. $i := (i+1) \bmod 256$
    b. $j := (j + s_i) \bmod 256$
    c. $t := s_i$
    d. $s_i := s_j$
    e. $s_j := t$
    f. $t := (s_i + s_j) \bmod 256$
    g. $b := s_t$
    h. o byte aleatório será o *b*

Algoritmo retirado de [*De Carvalho2000, p.98-100*].

Note que o vetor *S* muda à medida que se vão gerando bytes aleatórios. Isto contribui para a força do algoritmo.

Algoritmo Diffie-Hellman para troca de chaves

Esse é um outro algoritmo que eu acho importante mostrar aqui, porque ele é muito utilizado nos sistemas de troca de chaves.

Nós já vimos que os algoritmos são divididos em algoritmos que possuem a mesma chave para criptografar e descriptografar (simétrico) e aqueles que possuem chaves distintas (assimétrico). Quando possuem a mesma chave, se faz necessário um esquema em que o remetente possa combinar com o destinatário qual será a chave de desencriptação. Entretanto, se não houver um canal seguro para se passar à chave, tem que ser utilizado o mesmo canal (inseguro) para tal função. Para tanto, foram criados algoritmos para tornar essa passagem uma passagem segura. E o melhor e mais utilizado desses algoritmos é o Diffie-Hellman.

Criado por Whitfield Diffie e Martin Hellman em 1976 e publicado no artigo "*New Directions in Cryptografphy*", esse algoritmo foi fonte para vários produtos comerciais encontrados ainda hoje no mercado.

O objetivo desse algoritmo é habilitar dois usuários a trocar uma chave com segurança que será usada para a encriptação ou desencriptação.

O sucesso desse algoritmo é baseado na dificuldade de se calcular logaritmos discretos. De forma breve, nós podemos definir o logaritmo discreto da seguinte forma: primeiro definimos a raiz primária de um número primo p como um número cujas potências podem gerar todos os inteiros entre 1 e p-1. Ou seja, se a é uma raiz primária de um número primo p, então os números.

$$a \bmod p, a^2 \bmod p, ..., a^{p-1} \bmod p$$

são distintos e consiste nos inteiros de 1 a p-1 em alguma permutação.

Para qualquer *b* e uma raiz primária *a* do número primo *p*, pode-se encontrar um *i* tal que

$$b = a^i \bmod p \quad onde\ 0 \le i \le (p-1)$$

O expoente *i* é referenciado como o logaritmo discreto, ou índice, de *b* para a base *a*, mod *p*.

A partir daqui podemos definir o algoritmo de Diffie-Hellman para a troca de chaves.

Elementos públicos

q	Número primo
α	α<q e α uma raiz primitiva de q

Geração da base do usuário A

Selecione um X_A secreto	$X_A < q$
Calcule o Y_A	$Y_A = \alpha^{X_A} \bmod q$

Geração da base do usuário B

Selecione um X_B secreto	$X_B < q$
Calcule o Y_B	$Y_B = \alpha^{X_B} \bmod q$

Geração da chave secreta de A

$K = (Y_B)^{X_A} \bmod q$

Geração da chave secreta de B

$K = (Y_A)^{X_B} \bmod q$

Figura 2.3– Algoritmo de Diffie-Hellman

O número primo e a sua respectiva raiz primitiva são de conhecimento público, ou seja, pode trafegar livremente pelo canal inseguro. As bases, Y_A e Y_B, podem também circular livremente pelo canal inseguro.

As chaves encontradas pelos dois usuários serão idênticas por causa das propriedades mostradas a seguir:

$$K = (Y_B)^{X_A} \bmod q = (\alpha^{X_B} \bmod q)^{X_A} \bmod q =$$
$$(\alpha^{X_B})^{X_A} \bmod q = (\alpha^{X_A})^{X_B} \bmod q =$$
$$(\alpha^{X_A} \bmod q)^{X_B} \bmod q = (Y_A)^{X_B} \bmod q$$

Assim, os dois lados da comunicação conseguem trocar ou intercambiar suas chaves.

Desse algoritmo surgirá o protocoloco Oakley, importante componente do IPSec, que veremos no capítulo 7.

Resumo do Capítulo

O objetivo desse capítulo de apresentar a ciência da criptografia e falar sobre o principal algoritmo estudado durante esse curso, o RC4, foi alcançado. Mais a frente veremos que o RC4 é bastante utilizado e foi alvo de críticas de parte da comunidade científica e da totalidade da impressora especializada. Críticas essas, as quais discordamos.

Capítulo 3

Neste capítulo falaremos muito brevemente sobre as tecnologias, padrões ou protocolos que hoje temos disponíveis para o público consumidor. Em paralelo a essa monografia, o nosso colega Carlos Alberto Vieira Campos[4] vem fazendo um estudo muito mais avançado sobre esse tema.

Personal Area Network

As redes pessoais (PANs) são redes de dispositivos (Computadores, PDA, celular, wearables,...), na sua maioria móvel e sem fio, cujo diâmetro[5] é pequeno, algo até 45 metros no máximo.

As PANs podem ser formadas por dispositivos que utilizam uma infraestrutura, móvel ou não, ou dispositivos *ad hoc*, isto é, que não necessitam de infraestrutura.

IEEE802.11b (WLAN)

Este é um padrão projetado pelo IEEE (*Institute of Electrical and Eletronics Engineers*). E é um dos mais recentes padrões de redes sem fio. Ele tem as seguintes características

- Teoricamente, pode alcançar até 11 Mbps de velocidade, mas estudos dizem que ele aceita até 8Mbps.

- Tem conexão peer-to-peer ou baseada num ponto fixo de acesso.
- Opera na freqüência de 2,4GHz.
- Funciona com os dois métodos de espalhamento na freqüência: O FHSS (*Frequency Hopping Spread Spectrum*) e o DSSS (*Direct Sequence Spread Spectrum*). Além disso, funciona com infravermelho, que não utiliza nenhum desses dois métodos.
- Utiliza o protocolo de acesso CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*)
- Utiliza o WEP para autenticação e privacidade.

HiperLAN

O HiperLAN (*High Performance Radio LAN*) é um protocolo possui 4 tipos, todos criados pelo ETSI (*European Telecommunications Standarts Institute*).

Tipo	Descrição	Diâmetro	Velocidade
1	WLAN	50m	23,5 Mbps
2	ATM fixas	50-100m	> 20 Mbps
3	WLL	5000m	> 20 Mbps
4	WATM p-to-p	150m	155 Mbps

Tabela 3.1– Tipos de HypeLAN

Possui as seguintes características:

- Opera em faixas altas de velocidade.
- Opera na faixa de freqüência de 5 GHz.
- Implementa quesitos de QoS.
- Autenticação e Criptografia através do WEP.

[4] carlosvc@cos.ufrj.br
[5] Diâmetro é a menor distância entre os dois nós mais afastados.

HomeRF

Esse padrão foi desenvolvido pelo HomeRF Working Group, que é presidido pela Proxim. É um protocolo totalmente voltado para redes domésticas, operando em redes *ad-hocs*, ou podem operar em redes estruturadas, como ponto de acesso central, voltado para pequenas redes profissionais. Se for necessário o tráfego de voz, então é obrigatório montar a rede HomeRF sobre uma estrutura que permita um roteamento ligeiro. Seguem as características principais:

- Opera na faixa de freqüência de 2,4 GHz.
- Utiliza método de espalhamento de freqüência FHSS.
- Opera em distâncias até 45 metros.
- Autenticação e Criptografia através do WEP.

Bluetooth

Embora não seja o protocolo mais recente, é o que mais está em moda, tornando-se coqueluche no mercado de redes sem fio, principalmente nas redes pessoais. Essa notoriedade se deve principalmente ao baixo custo dos dispositivos que funcionam sobre Bluetooth. Ele foi criado pela Bluetooth Working Group. Seguem as características:

- Velocidade até 1Mbps.
- Opera no espectro de freqüência de 2,4GHz.
- Utiliza o método FHSS.
- Autenticação e Criptografia através do WEP.

Resumo

Como eu disse no início, o tema desse capítulo é objeto de estudo mais aprofundado do mestrando Carlos Alberto, eu só o introduzi para abrir discussão para o próximo capítulo. O que se pode observar nesses protocolos de redes sem fios é que todos utilizam o WEP para criptografar e autenticar nas mensagem que trafegam na rede.

Os próximos 3 capítulos tratarão mais especialmente do WEP, da forma como é implementados em um desses padrões de rede, o IEEE 802.11b, suas fragilidades e sugestões para melhor proteger os dados que trafegam nesse protocolo. O nosso estudo a partir daqui foi totalmente orientado ao WEP sobre o IEEE 802.11b.

Capítulo 4

As redes sem fio, como a IEEE 802.11b, possuem um conjunto adicional de elementos de segurança, chamado WEP, que não está disponível no mundo cabeado.

O WEP foi construído originalmente para atender as seguintes necessidades:

- **Grande confiabilidade**
- **Autosincronização:** Os clientes saem freqüentemente da área de cobertura.
- **Eficiência computacional:** O WEP foi construído para funcionar tanto em hardware quanto em software.
- **Exportabilidade:** Ele pode ser usado tanto nos padrões Americanos, quanto no dos outros países.
- **Opcionalidade:** O WEP não deve ser de uso obrigatório para manter compatibilidades com outros padrões.

O WEP utiliza a mesma chave para encriptar e desencriptar os pacotes. Veja como o algoritmo de encriptação do WEP funciona:

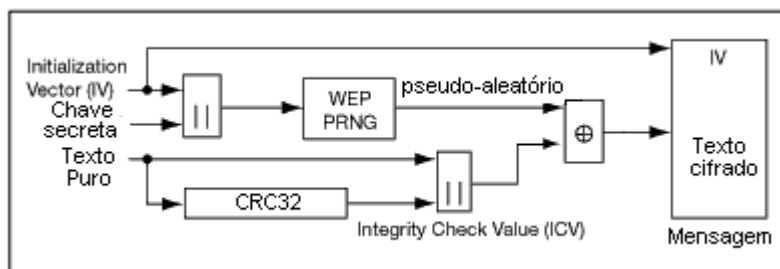



Figura 4.1– Esquema do WEP

Dois processos são aplicados sobre o texto puro. Um deles é o processo de encriptação e o outro é um processo que visa proteger quanto uma alteração não autorizada no texto durante a transmissão.

A chave secreta tem 40 bits e é concatenada com um Vetor de Inicialização (IV) de 24 bits, formando a chave composta que será responsável por chegar a string pseudo-aleatória de bits. A chave composta é inserida no algoritmo de PRNG (*Pseudo-random Number Generator*). O PRNG é baseado no algoritmo RC4 (*Ron's Cipher 4*), que vimos no capítulo 2 desse trabalho. A saída do algoritmo PRNG é uma seqüência pseudo-aleatória de bits, baseada na chave composta. Esta saída é utilizada para encriptar o texto puro através de uma operação binária de XOR. O resultado da encriptação é exatamente do tamanho do texto puro. A este resultado é concatenado, no início do pacote, o vetor de i-nicialização, e no final do pacote, 4 bytes (32 bits) resulta-do de um processo de ICV (*integrity check value*). O algoritmo de ICV é o CRC32. Esse conjunto, texto encripta-do, IV e ICV são enviados pelo canal inseguro. O CRC32 é utilizado para proteger os dados contra uma modificação não autorizada.

A estação destino, que de antemão já sabe o valor da chave secreta, usa o IV que vem no inicio do pacote para criar a mesma string gerada pelo PRNG e desencriptar o

texto cifrado. Então ele roda o CRC32 sobre esse texto desencriptado e recebe um novo valor de ICV. Ele compara esse novo valor de ICV com o valor que veio no final do pacote transmitido. Se os valores forem diferentes, o pacote é descartado, pois se tem certeza que a sua integridade foi quebrada.

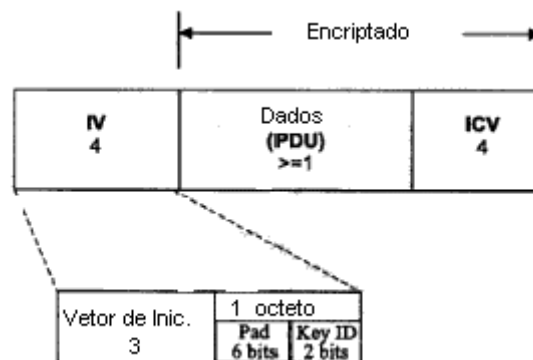



Figura 4.2– Esquema do pacote cifrado que é transmitido pelo canal inseguro

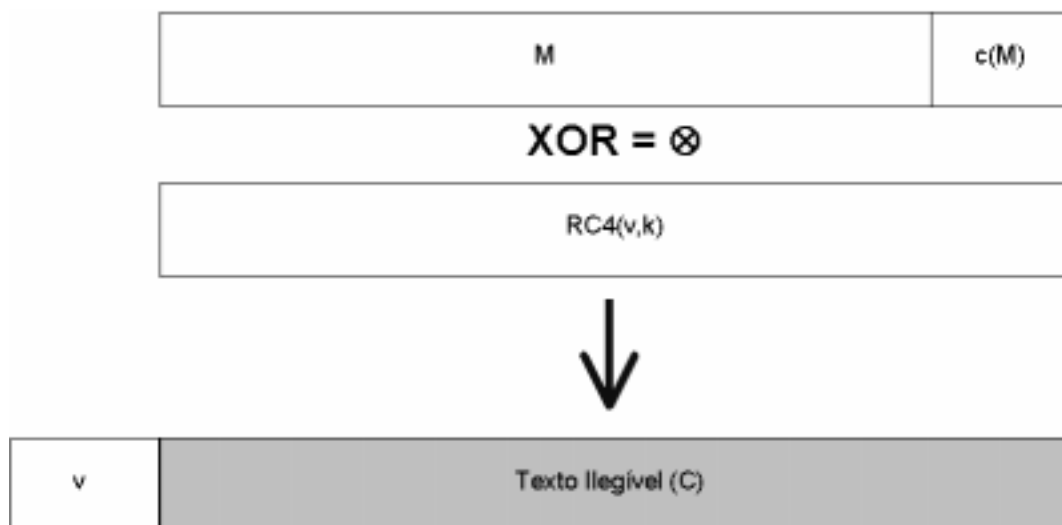



Figura 4.3– Operação lógica de Ou-exclusivo para encriptação.

No WEP, a mesma chave que é utilizada para encriptar e desencriptar é também utilizada para autenticar uma estação. Ter a mesma chave para encriptar e autenticar é considerado um risco de segurança. Existe também um método onde as estações que o ponto de acesso pode utilizar o WEP sozinho sem compartilhar a autenticação de chaves, essencialmente utilizando o WEP como encriptador.

Então, existe dois tipos de autenticação no IEEE 802.11:

- *Open system authentication*: Esse é o serviço de autenticação padrão. Não possui autenticação.
- *Shared key authentication:* Envolve uma chave secreta compartilhada para autenticar a estação a ponto de acesso.

Na *open system authentication* a estação pode associar com qualquer ponto de acesso e escutar todos os dados que são enviados sem encriptação. Isso é usado quando a facilidade de conexão é o principal objetivo, que o administrador está pouco preocupado com segurança.

A *shared key authentication* provê um nível mais alto de autenticação. A chave secreta reside em cada estação. O protocolo 802.11 não especifica como se distribui as chaves entre as estações.

O PRNG (RC4) é o componente mais crítico do processo WEP, já que é o real responsável pela encriptação. O vetor de inicialização estende a vida da chave secreta e provê a auto-sincronização do algoritmo. A chave secreta continua constante e o IV se altera periodicamente. O IV pode variar a cada envio de pacote.

Esse sistema é tão simples como parece, e nos próximo capítulo veremos que isso é uma das suas fraqueza e também uma das suas forças.

Capítulo 5

Vamos descrever como o WEP foi quebrado e como o IEEE 802.11b tornou-se totalmente inseguro no ano de 2001.

Antes de começar a ler esse capítulo é importante que você tenha lido o capítulo 2 e 4, pois citaremos os detalhes do algoritmo RC4 e a forma como ele é implementado no IEEE 802.11b.

Em julho de 2001, Nikita Borisov, Ian Goldberg e David Wagner, todos da Universidade da Califórnia (Berkeley), apresentaram na *International Conference on Mobile Computing and Networking*, acontecido em Roma, o seu artigo chamado *Intercepting Mobile Communications: The Insecurity of 802.11*. Esse artigo descreve as fraquezas do protocolo WEP, e são essas fraquezas que vamos apresentar a seguir.

Reutilização do vetor de inicialização

O vetor de inicialização no WEP tem 24 bits, e junto com a chave, é responsável por gerar a cadeia pseudo-aleatória (veja figura 4.3) que encripta o texto legível. O primeiro problema no WEP é justamente o tamanho desse IV que é muito pequeno. No caso extremo, esse IV é alterado a cada pacote enviado, começando no zero e indo até o valor máximo 2^{24}-1. Podemos calcular quanto tempo vai

demorar para esse *IV* voltar a assumir o valor 0 novamente: imagine uma conexão cuja banda seja de 5Mbits/s (o máximo no IEEE 802.11 é 11 Mbits/s, conforme já vimos).

$$(5Mbits/8)*1500 \cong 416 pac/s$$
$$2^{24} pac/416 \cong 40.329 seg \ ou \ 11h12m$$

Em suma, no caso mais extremo, numa conexão de 5Mbits/seg, o *IV* voltará a assumir o mesmo valor em menos de meio dia. Se a implementação assumir que o *IV* terá valores aleatórios teremos a repetição de um *IV* em menos tempo. E é a partir dessa repetição de *IV* que o WEP pode ser quebrado. A chave *K* é fixa, e foi configurada nos clientes que estão se comunicando, logo o par *<K,IV>* repetir-se-á sempre que o *IV* se repetir. E sempre que eles se repetirem, gerarão a mesma string pseudo-aleatória, que iremos referenciar como *RC4(K,IV)*.

Imagine dois textos legíveis distintos P_1 e P_2, que são criptografados através da mesma cadeia pseudo-aleatória *RC4(K,IV)* em C_1 e C_2.

$$C_1 = P_1 \otimes RC4(K,IV)$$
$$C_2 = P_2 \otimes RC4(K,IV)$$
$$C_1 \otimes C_2 = (P_1 \otimes RC4(K,IV)) \otimes (P_2 \otimes RC4(K,IV)) = P_1 \otimes P_2$$

Pelas propriedades do XOR (ou-exclusivo), visto no capítulo 2, podemos dizer que de posse de dois textos criptografados e um texto legível é possível descobrir o outro texto legível, pois:

$$C_1 \otimes C_2 \otimes P_1 = P_1 \otimes P_2 \otimes P_1 = P_2$$

E existem certos pacotes que tem o seu valor conhecido, alguns pacotes que possuem trechos conhecidos, como aqueles que pedem a chave do usuário; esses possuem

a palavra *password* e isso é de conhecimento geral. A partir de cada pacote novo descoberto, fica mais fácil descobrir outros, até que é possível conhecer todas as 2^{24} strings pseudo-aleatórias e todas os possíveis valores para o *IV*.

Gerenciamento de chaves

O padrão IEEE 802.11 não especifica como deve ser a distribuição das chaves. Ele é baseado num mecanismo externo de distribuição global da chave em um vetor de 4 chaves. Cada mensagem contém um campo de identificação de chave para especificar o índice do vetor da chave que está sendo usada. Na prática, a maioria das instalações utiliza a mesma chave para todos os dispositivos.

Isso traz problemas profundos à segurança dessas instalações, uma vez que a chave é compartilhada com vários usuários, fica muito complicado manter o segredo. Alguns administradores de rede tentam amenizar o problema não revelando a chave secreta ao usuário final, configurando, eles mesmos, os dispositivos. Mas isso não traz a solução, pois as chaves continuam guardadas nos dispositivos remotos.

A reutilização de uma única chave por vários usuários também aumenta as chances da colisão[6] do *IV*. A chance de uma colisão aleatória aumenta proporcionalmente ao número de usuários.

Uma vez que a troca de chaves requer que cada usuário reconfigure o seu dispositivo, as atualizações dos drivers controladores dos cartões de rede (NIC) serão cada vez mais infreqüêntes. Na prática, a troca demorará meses

[6] Colisão, nesse contexto, significa a captura de dois pacotes que utilizaram a mesma string pseudo-aleatória para a criptografia.

ou anos para acontecer, dando mais tempo para os intrusos analisarem o tráfego.

CRC32 linear

Outra grande fraqueza do WEP é o seu algoritmo de garantia da integridade (ICV - *integrity check value*), que é o CRC32.

O CRC32 é linear, isto é, $c(x \otimes y) = c(x) \otimes c(y)$ para qualquer valor de *x* e *y*. Essa propriedade serve para qualquer tipo de algoritmo CRC.

Uma conseqüência dessa propriedade é a possibilidade de se fazer modificações controladas no pacote, sem que sejam detectadas por qualquer um dos dispositivos transmissores ou receptores. Veremos que é possível alterar o conteúdo dos pacotes apenas com o conhecimento da string de valores pseudo-aleatórios.

Vamos lembrar como é formado o texto criptografado *C*, que corresponde ao texto legível *P*.

$$C = RC4(IV, K) \otimes \langle M, c(M) \rangle$$

Vamos imaginar um outro texto criptografado, *C'*, que seja a imagem da encriptação de um outro texto legível, *M'*, onde $M' = M \otimes D$, onde *D* é a alteração controlada que se deseja fazer. Veja só o desenvolvimento da fórmula a seguir.

$$C' = RC4(IV, K) \otimes \langle M', c(M') \rangle$$
$$C' = RC4(IV, K) \otimes \langle M \otimes D, c(M \otimes D) \rangle$$
$$C' = RC4(IV, K) \otimes \langle M \otimes D, c(M) \otimes c(D) \rangle$$
$$C' = RC4(IV, K) \otimes \langle M, c(M) \rangle \otimes \langle D, c(D) \rangle$$
$$C' = C \otimes \langle D, c(D) \rangle$$

Ou seja, pode-se interceptar o pacote, fazer a alteração, corrigir o ICV, e a alteração não será detectada, pois o sistema de manutenção de integridade foi perfeitamente burlado.

No mesmo artigo, o pessoal de Berkeley mostra como inserir e remover pacotes, usando essa propriedade da linearidade do CRC e usando o fato de que esse algoritmo possui chave, ou seja, não há proteção contra alteração do valor do ICV.

Esse artigo recomenda a utilização de um algoritmo de *Hash* para substituição ao CRC, entre outras coisas.

Correlação dos bytes da chave

O segundo trabalho que foi apresentado no ano de 2001 foi o trabalho de Scott Fluhrer, Itsik Mantin e Adi Shamir, chamado "*Weaknesses in the Key Scheduling Algorithm of RC4*".

Esse artigo tem um conteúdo muito mais matemático do que o anterior e fala da correlação entre os bytes da string pseudo-aleatória gerado pelo algoritmo RC4 com a chave (chave do WEP concatenada ao vetor de inicialização).

Você deve se lembrar do capítulo 2, quando descrevemos o algoritmo RC4, a expansão da chave não era nada mais complexo do que uma simples permuta dos números de 0 a 255, seguindo uma ordem estipulada pela chave.

Fluhrer e seus colegas enfatizam o fato de que o RC4 ser um algoritmo de criptografia de fluxo, ou seja, a cada byte gerado é imediatamente utilizado, isso faz com que o byte a ser utilizado na operação de XOR somente dependa

das iterações anteriores, o resultado conquistado daqui por diante não tem mais efeito sobre esse byte.

Observando a segunda parte do RC4, o PRGA (*Pseudo-Random Generation Algorithm*), é possível ver que o primeiro byte gerado é formado pelo byte S[S[1]+S[S[1]]], onde S é o vetor de bytes numerados de 0 até 255 que foi permutado pela primeira parte do algoritmo RC4, o KSA (*Key Scheduling Algorithm*).

1. i←0
2. j←0
3. Seja t um byte
4. Enquanto forem necessários bytes b aleatórios faz-se:
    a. i ← (i+1) mod 256
    b. j ← (j+s_i) mod 256
    c. t←s_i
    d. s_i ← s_j
    e. s_j ←t
    f. t ← (s_i +s_j) mod 256
    g. b ← s_t
    h. O byte aleatório será b

Figura 5.1– Algoritmo de geração de bytes pseudo-aleatórios.

i=1; j=S[1]; t=S[1]+S[j]=S[1]+S[S[1]]; b=S[t]=S[S[1]+S[S[1]]]

Figura 5.2– Cadeia de bytes gerada pelo KSA

Com esse "chinês" básico feito na figura 5.2, podemos acompanhar o valor das variáveis do algoritmo descrito na figura 5.1, e comprovar que o primeiro byte a sair é formado S[S[1]+S[S[1]]]].

Na continuação do artigo, os autores mostram que se a mesma chave secreta do WEP for utilizada com vários vetores de inicialização distintos, e o invasor puder obter o primeiro byte gerado pelo RC4 para cada vetor de iniciali-

zação, é possível reconstruir a chave secreta sem muito sacrifício.

Softwares

A partir do trabalho de Fluhrer et al, Stubblefield, Ioannidis e Rubin escreveram um artigo intitulado “Using the Fluhrer, Marton, and Shamir Attack to Break WEP”, e posteriormente foram criados dois softwares, o AirSnort e o WepCrack, que garantem quebrar o WEP com chaves de 40 bits de tamanho em 15 minutos.

O mais surpreendente dessa notícia é que Stubblefield garante que esse tempo para quebrar o WEP tem escalabilidade linear, ou seja, uma chave de 104 bits de tamanho demoraria menos de 40 minutos para ser quebrado.

Esses dois softwares foram construídos na mesma época e ambos rodam sobre Linux.

Existe ainda um software, o NetStumbler, que tem a função de testar se o WEP está habilitado ou não. Esse software roda sobre o Windows e é capaz de acessar todos os pontos de acessos existentes na região, informando quais deles estão com o WEP habilitado. Se acoplado a um GPS, ele ainda consegue identificar a posição exata do ponto de acesso.

Em defesa de Rivest

Na verdade dos dois artigos publicados em 2001 criticam a forma como foi implementado o WEP, mas nenhum dos dois vê defeitos no RC4.

Em seu artigo “RSA Security Response to Weaknesses in Key Scheduling Algorithm of RC4”, Ron Rivest, o “pai” do

RC4, defende-se das acusações de Fluhrer, Mantin e Shamir.

Primeiro, o RC4 é um algoritmo de criptografia que pode ser encontrado em outros lugares senão o WEP. O RC4 é utilizado em banco de dados e em sistemas operacionais e até mesmo em outros protocolos de segurança, sem trazer ônus nenhum para esses produtos.

Segundo, a forma como são escolhidos os vetores de inicialização, fazendo com que a chave do RC4 passe a ser um valor quase fixo que varie somente os últimos 24 bits, não é de responsabilidade do algoritmo RC4. Como também não é sua responsabilidade a utilização do fraco método de integridade.

Rivest termina sugerindo modificações no WEP para incrementar a segurança nesse protocolo. Uma das sugestões é a substituição do CRC32 por um algoritmo de hash, como o MD5 e o SHA1.

Capítulo 6

Esse capítulo será destinado as experiências feitas nos Estados Unidos pela revista ExtremeTech (http://www.extremetech.com), nas redes sem fio que seguem o padrão Wi-Fi.

Até agora falamos das falhas do WEP, neste capítulo veremos que muita gente não o utiliza para se defender. O cenário é alarmante.

Craig Ellison escreveu o artigo intitulado “Exploiting and Protecting 802.11b Wireless Networks”. Esse artigo é de fácil leitura e é recomendo para todos. Ele pode ser encontrado no *site* da ExtremeTech na internet.

As redes baseadas no protocolo IEEE 802.11b cresceram bastante em 2001, graças à brusca queda nos preços dos equipamentos. Nos EUA, o preço FOB para um cartão PCMCIA para um *notebook* está abaixo dos 100 dólares, e um ponto de acesso está na faixa dos 150 dólares. A fácil instalação de uma rede neste padrão também auxiliou para esse rápido crescimento das redes sem fios.

A primeira coisa observada é que os pontos de acesso estão instalados, na maioria das vezes, atrás do *firewall*. Ou seja, nesses casos os administradores só conseguem vislumbrar a hipótese de uma invasão através da rede cabeada, não se protegendo da invasão através da antena. As

vezes, o administrador ignora a existência de uma porta de entrada através do ponto de acesso. As redes sem fios são tão fáceis de serem instaladas que muitas das vezes os pontos de acessos são instalados sem a orientação, concordância ou conhecimento do departamento de informática ou CPD.

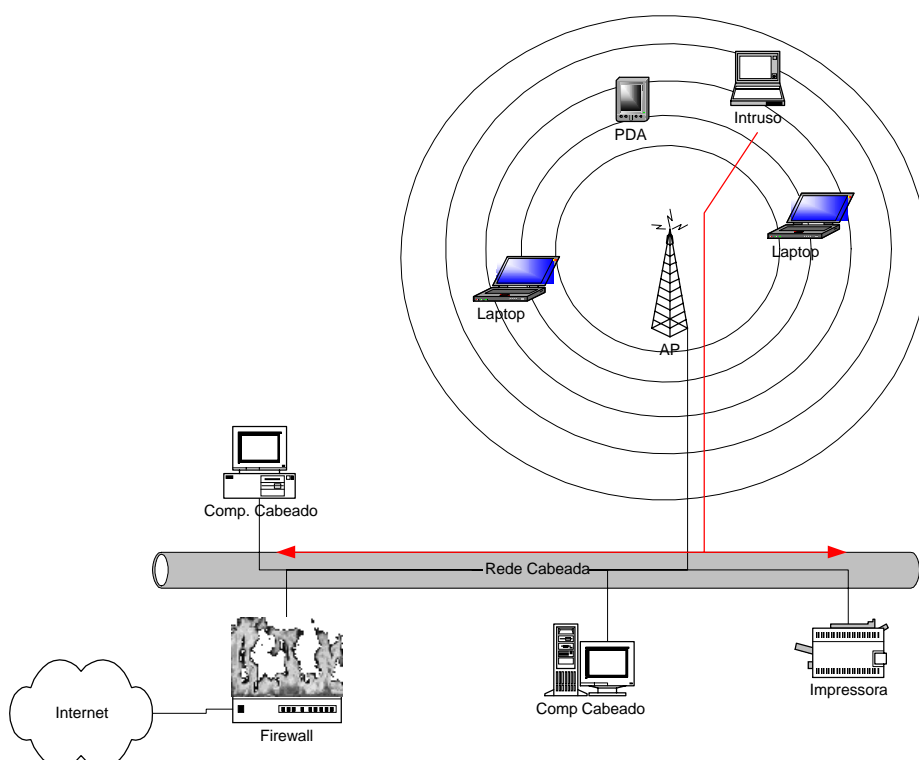



Figura 6.1– Diagrama de uma rede que tem o ponto de acesso atrás do *firewall*.

Faca de dois gumes

Toda padronização é uma faca de dois gumes: ao mesmo tempo em que facilita a compatibilidade de componentes de distintos fabricantes, poda boas isoladas idéias de melhoria.

Grande parte do crescimento dos produtos baseados no protocolo IEEE 802.11b deve-se ao trabalho da WECA (*Wireless Ethernet Compatibility Alliance*). WECA desenvolveu um protocolo de interoperacionalidade, chamado Wi-Fi (*wireless fidelity*), e todos os produtos que recebem o ca-

rimbo de compatibilidade Wi-Fi passam por uma grande bateria de testes de compatibilidade. O objetivo da WECA é garantir interoperacionalidade e facilidade de uso sem se preocupar com a segurança. Quando uma pessoa instala um componente com certificação Wi-Fi e tem garantia que esse componente funcionará com qualquer outro componente que tem a mesma certificação.

Esse é o principal problema, pois não cria nenhuma dificuldade para o invasor. Como o seu ponto de acesso é certificado pela WECA, o equipamento do invasor, que também será certificado, não terá nenhuma dificuldade para se comunicar com o seu ponto de acesso.

Teste do WEP

Em agosto de 2001 foi lançado na internet o programa chamado AirSnort, que é um programa que roda sobre o Linux como a versão 2.4 de kernel e placas de rede baseadas em Prismas. Esse programa é capaz de determinar a chave do protocolo WEP em segundos, depois de escutar algo entre 100 Mb e 1 Gb de tráfego. Isso imaginando que estamos falando em um protocolo WEP baseado em chaves estáticas, conforme a maioria das implementações encontradas no mercado. O AirSnort foi feito baseado na teoria de Fluher, Mantin e Shamir.

Utilizando o NetStumbler, outro *shareware* disponível na internet, foi possível descobrir que muita gente não habilita o WEP nas suas transmissões. O NetStumbler identifica o sinal do padrão IEEE 802.11b e registra os endereço MAC no ponto de acesso, o nome da rede, o SSID, o nome do fabricante, canal, se o WEP está habilitado ou não, a força do sinal, e várias outras *flags*. Além disso, se um GPS, que exporte resultados no padrão

GPS, que exporte resultados no padrão da NMEA (*National Marine Electronics Associations*), a latitude e longitude do ponto de acesso também é registrado.

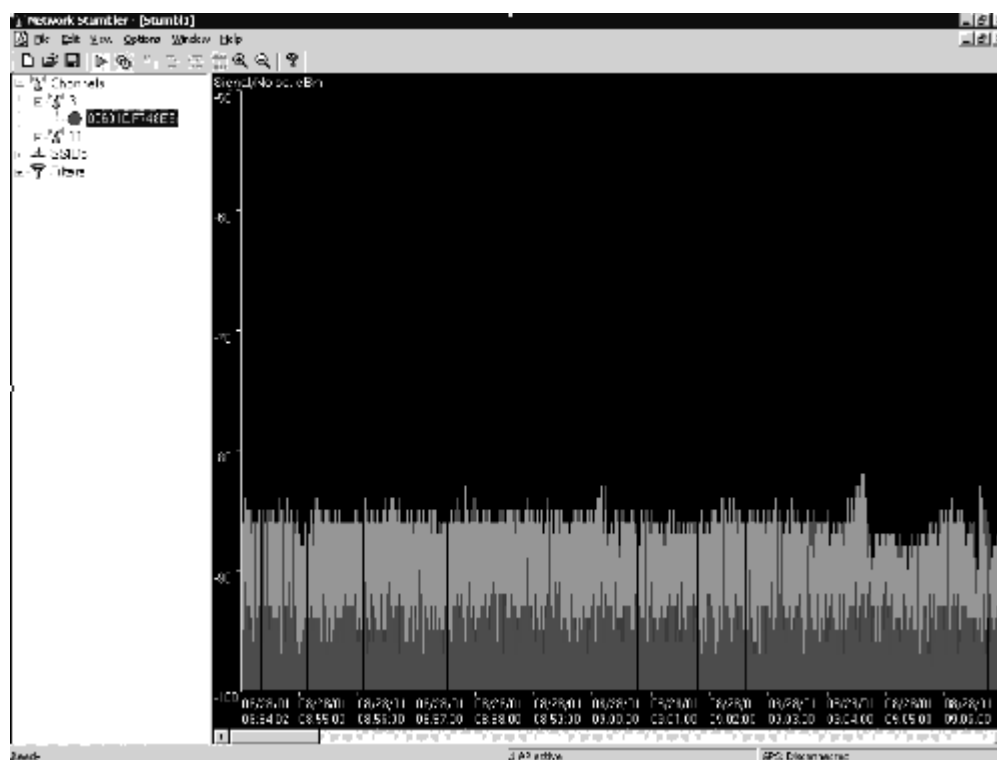



Figura 6.2– Imagem do NetStumbler

A maioria dos administradores assume que o sinal do IEEE 802.11b atravessa curtas distâncias – 30 m no máximo. Entretanto, esse sinal atravessa distâncias maiores, só que o sinal muito mais fraco e que são impossíveis de serem captadas pelas antenas internas dos cartões PCMCIA dos *notebooks*, entretanto quando o invasor utiliza uma antena externa, como uma 14dB yagi, por exemplo, o sinal do IEEE 802.11b pode ser captado por distâncias muito grandes.

Ellison, junto com Eric McIntyre, Christopher Hose e John Richey, armados com um cartão PCMCIA da Orinoco, uma antena de 14dB e uma antena onidirecional de 3dB, decidiram fazer uma experiência de resultados assustadores. Decidiram montar seus equipamentos e apontar as su-

as antenas ao acaso, com o objetivo de encontrar redes sem fios.

A primeira parada foi num terraço de um prédio na ilha de Manhattam. Em poucos minutos, eles já haviam encontrado 61 pontos de acessos ao redor daquele terraço. 79% dos pontos de acesso, ou seja, 48, não haviam habilitado o padrão WEP. Eles acessaram um dos pontos de acesso escolhido ao acaso, associado a esse ponto de acesso imediatamente receberam um endereço IP do servidor DHCP da rede em questão. Apontando o *browser* para o endereço IP do roteador (endereço do *gateway* que veio do DHCP), eles tiveram acesso a todas as informações do mesmo, uma vez que o administrador daquela rede não havia alterado a senha *default*.

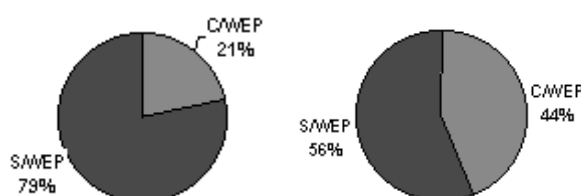



Figura 6.3– Resultado na Ilha de Manhattam e no Sillicon Valley

Mantendo a sua rede sem fio segura

O texto da ExtremeTech dá algumas sugestões que visam aumentar a segurança da sua rede sem fio. Na verdade essa lista de sugestões se aplica para todos os casos, sejam redes sem ou com fios.

1. Habilite o WEP. Como já vimos o WEP é frágil, mas ao mesmo tempo é uma barreira a mais no sistema de segurança.
2. Altere o SSID default dos produtos de rede. SSID é um identificador de grupos de redes. Para se juntar a uma rede, o novo dispositivo terá que

conhecer previamente o número do SSID, que é configurado no ponto de acesso, para se juntar ao resto dos dispositivos. Mantendo esse valor default fica mais fácil para o invasor entrar na rede.

3. Não coloque o SSID como nome da empresa, de divisões ou departamentos.
4. Não coloque o SSI como nome de ruas ou logradouros.
5. Se o ponto de acesso suporta broadcast SSID, desabilite essa opção.
6. Troque a senha default dos pontos de acessos e dos roteadores. Essas senhas são de conhecimento de todos os hackers.
7. Tente colocar o ponto de acesso no centro da empresa. Diminui a área de abrangência do sinal para fora da empresa.
8. Como administrador você deve repetir esse teste periodicamente na sua empresa a procura de pontos de acessos novos que você não tenha sido informado.
9. Aponte o equipamento notebook com o Netstumbler para fora da empresa para procurar se tem alguém lendo os sinais que transitam na sua rede.
10. Muitos pontos de acessos permitem que você controle o acesso a ele baseado no endereço MAC dos dispositivos clientes. Crie uma tabela de endereços MAC que possam acessar aquele

ponto de acesso. E mantenha essa tabela atualizada.

11. Utilize um nível extra de autenticação, como o RADIUS, por exemplo, antes de permitir uma associação de um dispositivo novo ao seu ponto de acesso. Muitas implementações já trazer esse nível de autenticação dentro do protocolo IEEE 802.11b, como veremos no próximo capítulo.

12. Pense em criar uma subrede específica para os dispositivos móveis, e disponibilizar um servidor DHCP só para essa sub-rede.

13. Não compre pontos de acesso ou dispositivos móveis que só utilizem WEP com chave de tamanho 40 bits.

14. Somente compre pontos de acessos com memória flash. Há um grande número de pesquisas na área de segurança nesse momento e você vai querer fazer um upgrade de software no futuro.

Para finalizar, você pode querer mais ou menos segurança na sua rede, mas com certeza quererá o mínimo, o básico, que é, mesmo com as suas vulnerabilidades, o WEP. Por isso habilite-o.

O WEP é fraco mesmo com uma chave de 104 bits. Nós vimos que não importa muito o tamanho da chave, só demorará mais ou menos tempo para quebrá-la, entretanto com essa pequena barreira você já difere um hacker novato de um curioso, o que já diminui muito o número de invasores na sua rede.

Capítulo 7

A primeira coisa a se pensar depois de lermos os últimos capítulos é que tudo está perdido. Até então se imagina que todo mundo está trabalhando contra o WEP e que não interessa a ninguém termos uma rede segura. Mas isso não é verdade. Já existem soluções que deixam a sua rede segura, e tem muita gente trabalhando para melhorar a segurança da sua rede, e não é preciso lhe dar outros exemplos de pessoas interessadas em segurança depois que eu citar as empresas de cartão de crédito.

Começando esse capítulo, vamos mostrar como a Cisco[7] resolveu o problema que seus produtos teriam com segurança se oferecessem somente o WEP, nada além.

Soluções CISCO

Autenticação Mútua

Os produtos Cisco para redes sem fio, conhecidos como *Cisco Aironet Wireless* oferecem um serviço de autenticação mútua. Isso consiste no ato da autenticação do cliente no ponto de acesso e o ponto de acesso no cliente. A Cisco criou o protocolo de autenticação EAP para assegurar a autenticação mútua entre o cliente e o servidor RADIUS (*Access Control Server 2000 v.2.6*)

[7] A Cisco foi escolhida por ser a empresa que disponibilizou o maior número de informações na internet, não tenho preferência por nenhum dos fabricantes.

Derivação da chave secreta

Originalmente o WEP utiliza a chave secreta para encriptar e desencriptar, e também para a autenticação. Os produtos Cisco não utilizam a chave secreta para autenticar, ao invés disso, eles utilizam uma chave derivada para essa mútua autenticação.

Chaves do WEP escolhidas dinamicamente

Como falamos anteriormente a chave do WEP costuma ser fixa, seja porque não é política da empresa trocar essas chaves, ou seja, porque o usuário é remoto e não tem conhecimento para trocar essa senha, sem auxílio do administrador. A Cisco oferece em seus produtos um sistema para troca das chaves a cada novo usuário e a cada nova conexão. Se o mesmo usuário tentar fazer uma nova conexão este receberá uma nova chave secreta.

Assim, a Cisco impede ou dificulta que o invasor fique escutando strings aleatórias geradas pela mesma chave por muito tempo.

Política de reautenticação

A política de reautenticação é forçar o usuário depois de um certo tempo, uma nova autenticação, e a nova autenticação determinará uma nova chave secreta, assim, mesmo que não se troque usuário e nem se troque uma sessão (como num processo de FTP), o cliente será obrigado a encriptar e desencriptar com outra chave secreta.

Alteração do Vetor de Inicialização

Como todas as implementações, os produtos da Cisco também incrementam o Vetor de Inicialização a cada pacote enviado. A diferença é que o vetor de inicialização começa-

rá a cada sessão a contagem a partir de um número escolhido aleatoriamente e não do zero como é em outras implementações.

Outros fabricantes

Eu não encontrei informações na internet a respeito das soluções encontradas por outros fabricantes. Repito, só por este motivo essa monografia cita a Cisco. Nós não preferimos nenhum fabricante em especial, preferimos sempre a melhor solução.

IP Security

Uma das melhores soluções é o IP Seguro. Entre as principais vantagens desta solução está no fato de que ela é transparente para a camada de aplicação e para o usuário. Fazendo com que, desta forma, não haja necessidade de alterarmos código de nossas aplicações nem precisemos de treinamento extra para os nossos usuários. Nos próximos parágrafos vamos falar um pouco sobre esta solução, tendo em mente que esse assunto foi discutido na monografia do Ismael Mariano (2000). É desta monografia e do livro do Willian Stallings que eu vou tirar a maioria dos assuntos.

Aspectos gerais do IPSec

O IPSec foi desenvolvido pelo IETF (*Internet Engineering Task Force*). Ele pretende substituir as vulnerabilidades do TCP/IP através da especificação dos seguintes serviços de segurança:

1. Controle de acesso
2. Integridade de pacotes
3. Autenticação da origem
4. Privacidade dos pacotes

5. Privacidade em fluxo de pacotes
6. Proteção de replays

O IPSec é de uso mandatário no IPV6.

Componentes do IPSec

O IPSec é composto por protocolos que são executados pelos nós da rede que se utilizam os seus serviços de segurança. Existem 3 protocolos:

- AH (*Authentication Header*)
- ESP (*Encapsulating Security Payload*)
- IKE (*Internet Key Exchange*)

O AH é o responsável pela autenticação, garantia de integridade e o combate ao *replay*. O ESP provê os serviços de criptografia e, opcionalmente, autenticação e anti-replay. O IKE é um protocolo híbrido, formado pelo ISAKMP (*Internet Key Management Protocol*) e pelo Oakley[8], e ele é responsável por gerar um meio seguro para que haja a troca de chaves na rede.

A operação de aplicar um determinado algoritmo de criptografia num pacote é chamada no IPSec de transformação. Durante a configuração de uma conexão que usa o IPSec para comunicar-se podemos definir uma ou mais transformações.

Todo o tráfego de uma comunicação via IPSec é executado sob o domínio de uma Security Association (SA) que é uma entidade peer-to-peer e simplex responsável por todas as informações de controle da sessão IPSec entre dois nós.

---

[8] Baseado na idéia de Diffie-Hellman

Por fim, temos os nós propriamente ditos que são os reais responsáveis pela inserção e/ou encaminhamento dos pacotes na rede. São eles que executam o software /hardware que implementa o IPSec. Existem dois tipos de nós: os *Security Gateway* (SG) e os *End Station IPSec*. Os SG disponibilizam os serviços de segurança para toda a rede (roteadores ou firewalls), enquanto os *End Station* fazem a segurança fim-a-fim entre os parceiros.

Para seu funcionamento o IPSec define várias estruturas de dados que são armazenadas em cada nó da rede que execute o IPSec. Este conjunto de dados forma dois bancos de dados a saber: O SPD (*Security Policy Database*) e o SAD (*Security Association Database*).

O SPD é composto por um conjunto de regras que determinam como processar os pacotes que chegam numa interface.

O SAD é composto por uma ou mais SA e armazena os parâmetros de cada uma delas. Ele é um banco de dados dinâmico, ou seja, suas entradas são excluídas após o termino da SA correspondente.

Uma SA identifica somente uma associação unidirecional entre dois nós com IPSec. Se a comunicação entre os dois nós for bilateral haverá duas SAs, uma de ida e outra de volta. Podem ter várias SAs entre dois nós. Numa mesma SA trafega somente um protocolo: AH ou ESP.

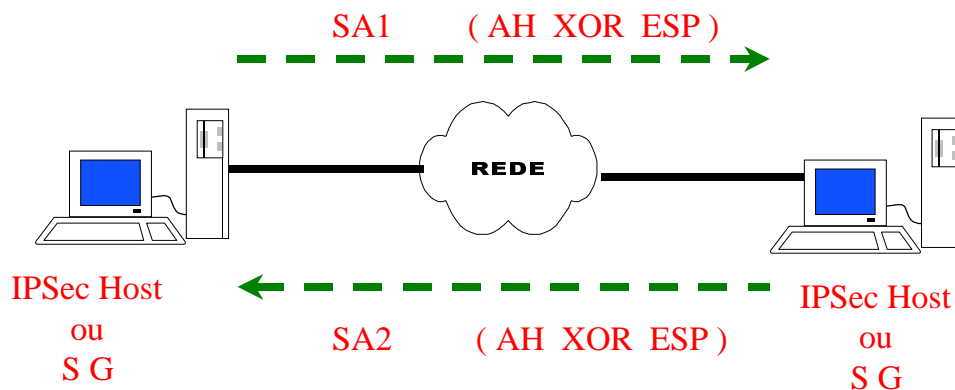
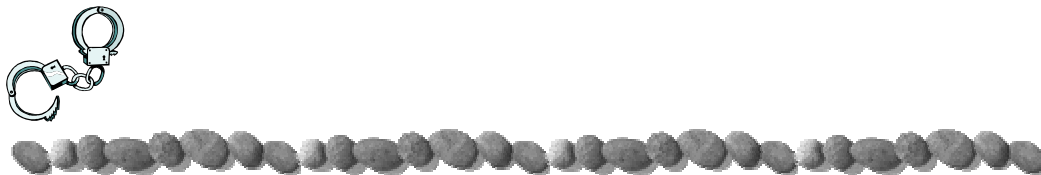



Figura 7.1 - SAs entre dois nós da rede

A SA é uma estrutura dinâmica e somente existe enquanto houver aquela conexão.

As SAs podem trabalhar em dois modos: Transporte e Túnel. O modo transporte é usado para prover segurança para comunicações fim-a-fim (cliente/servidor, duas estações de trabalho ou console de gerenciamento/dispositivo gerenciado). Nesse caso, o escopo de proteção do pacote restringe-se ao payload do IP (segmento TCP ou UDP e pacote ICMP).

O modo túnel é usado para prover segurança para comunicações entre redes ou entre uma estação e uma rede (tipicamente aplicações VPN[9]). Aqui o escopo de proteção é todo o pacote IP. Um novo cabeçalho é gerado e o cabeçalho original é incluso no payload do novo cabeçalho IP. O modo túnel é mandatário se uma das extremidades da conexão foi um SG.

A encriptação no ESP nunca segue um único algoritmo de criptografia (DES, RC5, IDEA, CAST, Blowfish,...). A variação é mais um modo de garantir segurança. Essa variação também é aleatória.

[9] Virtual Private Network – A ser definido ainda neste capítulo.

Desempenho

Logicamente com o IPSec o desempenho da comunicação cai um pouco. A fragmentação dos pacotes no IPSec tende a aumentar, pois haverá adição de cabeçalhos maiores do que no IP.

Como solução para os problemas de desempenho com redes em IPSec adotamos os seguintes procedimentos:

- Aumentar o poder de processamento de SG e Hosts;
- Realizar a compressão do IP payload através do protocolo IPPCP (*IP Payload Compression Protocol*);
- Realizar a criptografia e descriptografia em hardware específico.

Virtual Privacy Networks

Uma das melhores aplicações do IPSec são as VPNs. Na verdade VPN é um termo genérico para qualquer tecnologia que garanta comunicação segura sobre a internet pública.

A idéia da VPN é criar um túnel seguro entre os gateways para proteger os dados privados enquanto eles estão navegando pela internet, ou seja, enquanto eles estão trafegando por redes não confiáveis.

Figura 7.2 – Exemplo de uma VPN que liga dois roteadores.

Dentro do túnel, todos os dados, incluindo os cabeçalhos, são encriptados. A forma como esses dados serão encriptados, quais os algoritmos de encriptação e autenticação serão utilizados, dependerá do protocolo sobre o qual a VPN está se baseando. O IPSec não é a única solução para uma VPN, existe um protocolo chamado PPTP que também se presta para esse fim com muita qualidade.

As VPNs não servem somente para ligar gateways, elas também podem conectar dispositivos às redes. A seguir veremos os tipos de conexões que são suportadas pela VPN:

- Cliente-Rede: É quando um dispositivo isolado quer se conectar a uma rede. É utilizado por trabalhadores que não trabalham em um lugar fixo e necessitam de mobilidade. Se um vendedor está hospedado num hotel em outra cidade distante da sede da sua empresa e necessita fazer um *upload* da base de novos clientes com segurança ele pode fazer uma conexão local, na cidade onde se encontra, e utilizar a internet pública para chegar até os servidores da sua empresa. A VPN faz com que o trajeto do hotel até a empresa seja um túnel inviolável.
- Rede-Rede: A VPN é utilizada para unir duas redes. Independente da distância, há casos em que você precisa ligar dois escritórios. Normalmente essa conexão é feita através da rede de telefonia pública. A VPN faz com que haja segurança entre os dois roteadores.

- Intranet: Aqui a VPN é utilizada para que informações que são exclusivas de um ou mais departamentos, não seja acessadas por toda a empresa.
- Extranet: A VPN é utilizada para assegurar que as informações serão vistas somente pelos clientes e/ou fornecedores.

Conclusão

Pelo que nós vimos, 2001 foi um ano muito bom a comunicação sem fio. Sim, muito bom, o fato de o WEP ser quebrado duas vezes demonstrou ao mundo que o WEP está sendo estudado e que ele vai evoluir. Consideramos que só testando exaustivamente um sistema poderemos atestar que ele é confiável ou não. E mesmo quando ele é dito confiável não temos a segurança perfeita. Na verdade não existe essa segurança perfeita, qualquer sistema poderá ser violado, é só questão de tempo. As pesquisas que vêm sendo feitas são para conquistar novas soluções que substituam as anteriores antes de sua quebra.

Achamos que esse trabalho serviu para expor melhor o conteúdo do algoritmo RC4. O RC4 é um algoritmo simples e muito seguro, de modo nenhum ele foi o responsável pela quebra do WEP.

Em de 17 de dezembro de 2001, a RSA Security, em parceria com a Hifn uma empresa de infraestrutura de redes, anunciou o lançamento do RC4 Fast Packet Keying que é uma algoritmo para geração de chaves para o RC4. O Fast Packet Keying utiliza uma função *hash* para garantir que todo e cada pacote encriptado pelo WEP terá uma chave distinta.

Criar um padrão é uma tarefa muito complicada, ao mesmo tempo que a comunidade científica exige segurança e robustez, a indústria exige facilidade para a instalação. A industria quer padrões que proporcionem dispositivos *plug'n play*, mas não é só isso que os usuários existem dos seus dispositivos. Por mais tempo que leve, os usuários serão responsáveis por ditar a direção do mercado, pois eles serão prejudicados por uma instalação sem segurança.

O que vimos aqui se trata do conteúdo apresentado durante o curso de Tópicos Especiais em Redes Integradas Faixa Larga. Esse curso demorou oito semanas, ou um mês e meio, e isso não é tempo suficiente para se esgotar um assunto. Para começar a estudar esse assunto nós precisávamos dessa base inicial. Eu, particularmente, gostei muito de ler a respeito de criptografia, principalmente no livro do Routo TERADA. Gostaria de estudar mais a esse respeito, pois entendo que no futuro os sistemas de segurança confiarão na aleatoriedade como que se decidem quais dos bons e eficientes algoritmos de criptografia serão utilizados, sozinhos ou simultaneamente.

Apêndice A - Glossário

Ataque
Ato de violar ou tentar violar sistemas de segurança

Privacidade
Garantir que as informações não sejam acessadas por pessoas ou programas não autorizados.

Autenticidade
Possibilidade de identificar, sem equívocos, a autoria de determinada ação, ataque ou não.

Integridade
Impossibilidade de modificação, intencional ou não, de dados ou recursos

Back door
Um programa, deixado por um intruso, que permite futuro acesso à máquina alvo, sem a necessidade de autorização

Bug
Uma falha num programa. Um bug pode trazer vulnerabilidade ao sistema de segurança, mesmo que esse programa não faça parte do sistema.

Cavalo de Tróia, Trojan Horse
Uma aplicação que realiza alguma tarefa que compromete a segurança do sistema. Tem esse nome porque esse programa vem camuflado em um outro programa que atraia a atenção do usuário, como um joguinho ou uma animação.

CERT
Computer Emergency Response Team – Organização dedicada à segurança, seu propósito é socorrer redes que foram atacadas. Ex.: CAIS (Centro de Atendimento a Incidentes de Segurança) da RNP

Certificação
Serve para validade se um sistema está seguro. É feita por uma equipe especialista que procura falhas no sistema.

Crack
Programa utilizado para quebrar senhas

Cracker
Indivíduo que ataca sistemas de segurança com intenções criminosas.

Hacker
Indivíduo que ataca sistemas de segurança com intenções de diversão ou emoção. Em geral hackers não destroem os dados. Alguns dizem ter ética.

Engenharia Social
Técnica utilizada por intrusos para obter informações relevantes ao ataque, diretamente de pessoas.

Exploit
Programas utilizados por intrusos para explorar vulnerabilidades em determinados sistemas, conseguindo assim, acessos com maior privilégio

Firewall
Hardware e/ou Software que controla o fluxo de conexões que sai ou entra na rede.

Hacking
É o ato de hackear sistemas, não no sentido único de invadir, mas principalmente saber como funcionam e se possuem falhas.

Hijacking
É um seqüestro de uma sessão, geralmente TCP/IP. O seqüestro é uma forma de obter controle de uma conexão iniciada por um usuário legítimo. Ao interceptar essa conexão o hacker ou cracker por tomar o lugar o usuário legítimo. Essa conexão já passou pelo sistema de autenticidade.

Hole
Um buraco que deixa o sistema vulnerável.

IDS
Intrusion Detection System – É um sistema de deteção de intrusão, um software responsável por monitorar uma rede ou sistema e alertar sobre possíveis invasões.

Invasão
Caracteriza um ataque bem sucedido.

Lammer
É uma palavra que os hackers utilizam para identificar os indivíduos que pensam ser hackers.

Phreaking
São os hackers de telefonia convencional ou celular

Scanner
Ferramenta utilizada por hackers ou especialistas em segurança que serve para “varrer” uma máquina ou uma rede, em busca de portas abertas, informações ou serviços vulneráveis.

Script Kiddie
É um indivíduo que saiu do estágio de lammer, mas que só sabe utilizar scripts.

Sniffer
Ferramenta que serve para monitorar e gravar pacotes que trafegam pela rede.

Spoofing
É uma forma de manter uma conexão com uma máquina se fazendo passar por uma outra na qual ela confie.

Vírus
São programas que infectam outros programas e se multiplicam, na maioria das vezes podem causar danos aos sistemas infectados

Flood
Sobrecarga (em geral, de pacotes) causada por eventos não esperados que causam lentidão na rede.

Worm
Semelhante ao um vírus, mas difere pelo fato de não necessitar de um programa portador para se infectar.

Referências Bibliográficas

ARBAUGH, W.A.; SHANKAR, N.. WAN, Y.C.J. *Your 802.11 Wireless Network has No Clothes*. http://www.cs.umd.edu/~waa/wireless.html. 2001.

BARBOSA, André S.; DE MORAES, Luís Felipe M. *Curso de Sistemas de Segurança de Informação*. Rio de Janeiro: RAVEL/UFRJ. 2000.

BORISOV, Nikita; GOLDBERG, Ian; WAGNER, David. Intercepting Mobile Communications: The Insecurity of 802.11. *The Seventh Annual International Conference on Mobile Computing and Networking*. July 16-21, 2001, Rome, Italy. 2001.

CAMPELLO, R.S.; WEBER, R. Minicurso de Sistemas de Detecção de Intruso. In: *19º Simpósio Brasileiro de Redes de Computadores*. Florianópolis, 21 a 25 de maio de 2001.

DE CARVALHO, Daniel Balparda *Segurança de Dados com Criptografia: Métodos e Algoritmos*. Rio de Janeiro: Book Express. 2000. 253p.

DIFFIE, W.; HELLMAN, M. New Directions in Cryptography. *IEEE Transactions on Informations Theory*. November 1976.

ELLISON, C. Exploiting and Protecting 802.11b Wireless Networks. In: *ExtremeTech*. Sep. 4, 2001. 2001.

FLUHRER, S.; MANTIN, I.; SHAMIR, A. Weakness in the Key Scheduling Algorithm of RC4. Presented at *8th Annual Workshop on Selected Areas in Cryptography*. 2001. 23p.

MARIANO, Ismael da S. IPSec e DDoS, Aspectos de Segurança em Redes TCP/IP. *Seminário de Tópicos Especiais em Redes Integradas Faixa Larga*. Rio de Janeiro: COPPE/Sistemas. 2000.

MARTINS, Alessandro. Sistemas de Autenticação e Certificação. *Seminário de Tópicos Especiais em Redes Integradas Faixa Larga*. Rio de Janeiro: COPPE/Sistemas. 2000.

MATEUS, Geraldo Robson; LOUREIRO, Antônio A. F. Introdução à Computação Móvel. Rio de Janeiro: DCC/IM, COPPE/Sistemas, NCE/UFRJ, 1998.

McCLURE, Stuart; SCAMBRAY, Joel; KURTZ, George. *Hackers Expostos: Segredos e soluções para a segurança de redes*. New York: McGraw-Hill. 1999. Traduzido por ANTUNES, Álvaro. São Paulo: Makron Books. 2000. 469p.

MEREDITH, G. Securing the Wireless LAN. *Packet magazine*. v. 13. n. 3. p 74-77.

MILLER, Michael. *Descobrindo Bluetooth*. Rio de Janeiro: Campus. 2001.

NEGUS, K.J.; STEPHENS, A.P.; LANSFORD, J. *HomeRF: Wireless Networking for the Connected Home*. http://www.homerf.org. Last Access: 28-Nov-01. 2001.

ORMAN, H. K. *The OAKLEY Key Determination Protocol*. http://www.imib.med.tu-dresden.de/imib/Internet/Literatur/ISAKMP/draft-ietf-ipsec-oakley-02.txt. Last Access: 06-Dec-01.

Paulson, L.D. Exploring the Wireless LANscape. *Computer Magazine*. Outubro/2000. 2000.

PRAZERES, C.V.S.; MAIA Jr., G.B.; REIS JR., P.B. *Fundamentos Teóricos da Criptografia*. Salvador: Universidade de Salvador, Departamento de Informática. 2000.RIVEST, R. RSA Security Response to Weaknesses in Key Scheduling Algorithm of RC4. *RSA Security's Site*. http://www.rsasecurity.com/rsalabs/technotes/wep.html. Last Access: 13-Dec-2001. 2001.

SHANNON, C.E. A Mathematical Theory of Communication. *The Bell System Technical Journal*. v.27, p.379-423, 623-656,Julho,Outubro. 1948.

SOARES, L.F.G.; LEMOS, G.; COLCHER, S. *Redes de Computadores: das LANs, MANs e WANs às redes ATM. Rio de Janeiro*: Campus. 1995. 710p.

STALLINGS, W. *Cryptography and Network Security*. 2 ed. New Jersey: Prantice-Hall. 1998. 569p.

STUBBLEFIELD, A.; IOANNIDIS, J.; RUBIN, A.D. Using the Fluhrer, Mantin, and Shamir Attack to Break WEP. *ATT&T Labs Technical Report TD-4ZCPZZ*. Rev. 2. Aug. 21, 2001.

TANEMBAUM, Andrew S. *Redes de Computadores*. 1997.Tradução da Terceira Edição. Rio de Janeiro: Campus. 1944.

TERADA, Roto. *Segurança de Dados: Criptografia em Redes de Computadores*. São Paulo: Edgard Blücher. 2000.

VERISSIMO, Fernando. Quando os ex-empregados tornam-se hackers. In: *Portal Lockabit de Segurança*. http://www.lockabit.coppe.ufrj.br. 2001.

_________ Novo Dicionário Aurélio da Língua Portuguesa. FERREIRA, Aurélio Buarque de Holanda; J.E.M.M. Editores Ltda. Rio de Janeiro: Nova Fronteira. 2ed. 1986. 1862p.

Sobre o autor

Fernando Carlos Azeredo Verissimo é carioca, nascido em 1972. Formou-se em Bacharelado em Matemática modalidade Informática na Universidade do Estado do Rio de Janeiro, e vem trabalhando desde 1990 como Analista de Sistemas. Fez a sua pós-graduação em Gerência de Sistemas na Universidade Estácio de Sá até 2001. Também fez, mas não concluiu, um curso de Bacharelado em Estatística. Atualmente está cursando o Mestrado em Engenharia de Sistemas da COPPE/UFRJ. Desde 1995 trabalha como Gerente de Informática e Tecnologias na Academia Brasileira de Ciências. Email: verissimo@pobox.com

Índice

Índice de Figuras e Tabelas